عيد مبارك!

مجلة
مجتمع لينوكس العربي

مجلة تعنى بشؤون المصادر الحرة

العدد ٥ سبتمبر/ أكتوبر ٢٠٠٨
http://www.linuxac.org

إقرأ في داخل العدد:

* أنشئ خادم "فويب" مع OPENSER
* الحقوق المرفوعة: المثالية الواقعية
* التعامل مع الأراشيف والملفات المضغوطة في جنو / لينوكس
* "سايثون" : تجمع بين "سي" و"بايثون"
* طريقة إنشاء فيديو رسوم متحركة في جنو / لينوكس
* مقدمة إلى Rootkit
* الشبكات اللاسلكية وأساسيات حمايتها
* والعديد من المواضيع الجديدة والقيمة.

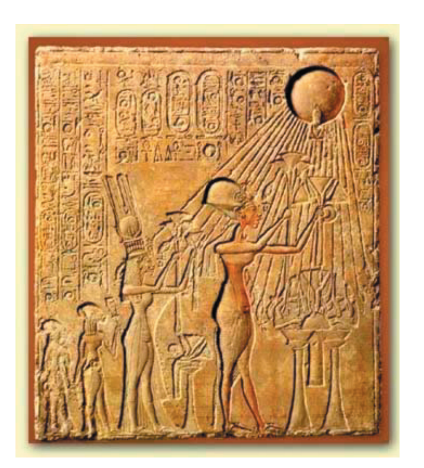
صورة الغلاف: joe shlabotnik

فهرس العدد

كلمة العدد ٣
أخبار متفرقة ٤
مقال بعنوان: الحقوق المرفوعة : المثالية الواقعية ٦
أنشئ خادم فويب مع OPENSER ٩
التعامل مع الأراشيف والملفات المضغوطة في جنو/لينوكس ٢٠
"سايثون" : تجمع بين "سي" و "بايثون" ٢٥
طريقة إنشاء فيديو رسوم متحركة باستخدام Inkspace و
ImageMagick و FFmpeg ٣١
الشبكات اللاسلكية وأساسيات حمايتها ٣٦
مقدمة إلى الـ RootKit ٣٩
تثبيت أوبن سوزه ١١ على ذاكرة حية خارجية Live USB ٤٥
أداة MSEC لإدارة حماية النظام على "ماندريفا" ٤٨
نتائج مسابقة قسم نفحات رمضانية في مجتمع لينوكس
العربي ٥٤
فريق عمل المجلة ٥٥

كلمة العدد

بسم الله الرحمن الرحيم



وتتحقق النبوءة! ... وتخرج من رحم التاريخ مكتملة الملامح ... تطير في سماء ملبدة بالغيوم السود ... متجهة نحو نور يتخلل تلك الغيوم المعتمة ... نحو الحرية ... نحو السمو ... إلى نقطة اللا رجوع!

كلا ليست تلك الكلمات من إحدى مسرحيات "ملاحم" التاريخ الأسطورية كما قد يبدو للوهلة الأولى ، فهي كلمات من الحاضر والواقع الذي نعيشه اليوم ، فالواقع الذي يبنى بسواعدنا العربية ، في مجتمعنا العربي ، هو ملحمة بحد ذاته ، وسيسجلها التاريخ في كتبه يوماً ما ، وسيقرأ تلك الكتب أبناء وأحفاد تلك السواعد ، التي جسدت تلك الملحمة.

قد يستغرب البعض أو قد يجدني أبالغ في الوصف قليلا ... حسناً ... ربما كثيراً ، ولكنني سأوضح لكم لم اخترتُ هذا الوصف كي أطلقه على ما نخوضه في أيامنا هذه وفي مجتمعنا العربي.

كما هو من المعروف لنا أن لكل ملحمة تاريخية بطل ما ، وملحمتنا هذه بها العديد من الأبطال ، ولكنهم أبطال حقيقيون لم تجسدهم مخيلة فيلسوف يوناني ما في حقبة ما قبل التاريخ الجديد ، فهم يعملون بجهد وتفانٍ ولا يطلبون شكراً ولا عرفاناً ، ويبذلون الكثير ويقدمون كل ما لديهم لخدمة الأمة وأبناء هذه الأمة ، وفوق ذاك وذاك كله فهم يعملون بالخفاء ومن وراء الكواليس لا يريدون أن تطالهم أضواء شهرة ولا يطلبون مجداً زائفاً ، فهل لدى "الأوديسة" أو "الإلياذة" مثل هؤلاء الأبطال؟ كلا ... لا أعتقد ذلك!

وهناك العديد من المتربصين لأولئك الأبطال الأسطوريون، والذين ينتظرون لحظة سقوط "جلجامش" الجديد ، والعديد من المنافقين والمتلونين ، والأعداء الظاهرين و المتخفين ، وهم أيضا موجودون في زمننا هذا ، و هم ليسوا من مخيلة ذاك الكاتب أو الفيلسوف الذي ذكرناه قبل قليل ، فهم واقع حزين ... بل واقع أليم ، كم هو مؤلم أن ترى ذلك من أبناء جلدتنا وأبناء لغتنا وأخوتنا في الدين والعرق والدم.

ولكن ... فكما كتب التاريخ ملاحم الماضي والحاضر ، من ملحمة "إنزو" ، وحتى ملحمة "الحرافيش" ، فسيكتب غيرها ، وسيبقى هناك أبطال، وسيبقى هناك .... أشباح أبطال!



رئيس التحرير

أخبار متفرقة

إعداد: أحمد عبد الرحمن



برنامج جمب يحلق من جديد في سماء لينُكس

بعد أشهر طويلة من التطوير صدرت أخيراً النسخة النهائية من برنامج جمب للصور النقطية وثنائية الأبعاد رقم ٢،٦ ، وكان ثمة تأجيل متكرر أحل بموعد صدور النسخة النهائية من البرنامج حتى صدرت النسخة الجديدة بتحديثات ثورية شملت أدواتها وشكل البرنامج وواجهته والمكتبات الرسومية التى يعتمد عليها والأهم من هذا إعتماد مكتبة جى إى جى إل (GEGL) والتى ستتيح دعم لأهم ما ينقص البرنامج حالياً مثل دعمه للعمق اللونى الأكبر ١٦ بت وكذا دعمها للنموذج اللونى المخصص للطباعة cmyk وغير هذا ومن المنتظر أن يكتمل دعم البرنامج لتلك المكتبة فى النسخة القادمة رقم ٢،٨ عام ٢٠٠٩
ولتحميل البرنامج والإطلاع على أهم ما حل به من تحديثات يرجى زيارة موقع البرنامج الرسمىwww.gimp.org
ولرؤية تقرير عربى عن النسخة الجديدة من البرنامج وما أتت به يرجى زيارة هذا الموضوع بمجتمع لينوكس العربى

http://www.linuxac.org/forum/showthread.php?t=15951

نسخة جديدة من برنامج بلندر

برنامج بلندر للرسوم ثلاثية الأبعاد الحر والمجانى يستعد هو الآخر فى الأيام القادمة لطرح نسخة جديدة تحمل رقم ٢،٤٨
ولا شك فى أن هذا البرنامج شمل طفرة كبيرة فى الشهور السابقة فقد شهد صدور عدة نسخ فى الشهور السابقة والعديد من الأعمال المنتجة بواسطته من قبل مطورى البرنامج والتى ساهمت فى تطويعه وتحديثه بصورة أكبر وهو الأسلوب الذى ينتهجه مطورى البرنامج منذ فترة خصوصاً بعد رعاية شركة صن ميكروسيستم للبرنامج
ولمعرفة أخبار البرنامج وتحميل نسخه الجديدة وكذا دروسة ومعرضه يرجى زيارة موقعه الرسمى
www.blender.org

ماندريفا لينُكس ٢٠٠٩: فاتنة توزيعات لينُكس تصدر فى حلة جديدة

التوزيعة الفرنسية العريقة ماندريفا صدرت نسخة جديدة منها متزامنة مع مرور عشر سنوات على أول ظهور لها ولزيارة موقع التوزيعة الشهيرة تفضلوا من هنا www.mandriva.com

ولمشاهدة تقرير عربى عن التوزيعة الجديدة تفضلوا بزيارة هذا الموضوع فى مجتمع لينوكس العربى

http://www.linuxac.org/forum/showthread.php?p=140672

مونو ٢,٠

أعلن فريق مشروع مونو الحر عن صدور الإصدارة الثانية والنهائية الجديدة من البرنامج ، و هو عبارة عن إعادة كتابة إطار العمل NET. من مايكروسوفت ، ليكون قابل للنقل على منصات الويندوز و اليونكس و ماكنتوش و اللينكس و غيرها من الأنظمة .
ولتحميل البرنامج ومتابعة جديدة يرجى زيارة موقعه الرسمى http://mono-project.com

وهنا تقرير عربى عن المشروع فى موقع وادى التقنية

http://www.itwadi.com/mono-2-0

موسوعة المعرفة

نجم عربى جديد فى سماء المعرفة الحرة .. تلك الموسوعة العربية التى اتطلقت فى عام ٢٠٠٧ تواصل مسيرتها المعرفية الخلاقة والحرة والقائمة على توثيق شتى مناحى المعرفة باللغة العربية بكل حيادية وبعيد عن قيود الغرب وصراعات اللغة والدين والتاريخ والتى ارتبطت بهم فى موسوعات أخرى شهيرة

يذكر أن مؤسس تلك الموسوعة هو الدكتور : نايل الشافعى صاحب ومدير إحدى شركات الاتصالات في الولايات المتحدة والمحاضر في معهد مساتشوستش للتكنولوجيا MIT والاستشاري للعديد من الهيئات الدولية والعالمية منها هيئة الاتصالات الفيدرالية الأمريكية FCC وبرنامج الأمم المتحدة الإنمائي
ولزيارة الموسوعة تفضلوا من هنا www.marefa.org

إطلاق موسوعة wiki مجتمع لينوكس العربي الجديدة

إنطلاقة جديدة لمجتمع لينوكس العربى ... فمع الجهد الكبير والمبذول من قبل مجلس إدارته ومراقبيه ومشرفيه وأعضاءه الكرام فى خدمة ورعاية المجتمع وحشد القوى الفكرية والعلمية لخدمة العلم والعلماء بتقديم ونشر العلم الخلاق والهادف بعيداً عن لغة التجارة والإحتكار الهادمة ، ومع هذا الجهد يعلن المجتمع عن إطلاق ويكى المجتمع www.linuxac.org/wik ليكون مُكملاً للمادة العلمية الكبيرة بمنتدى المجتمع وموثقاً لها بصورة مرتبة ودقيقة من قبل مجلس خاص لها تشرف عليه إدارة المجتمع .

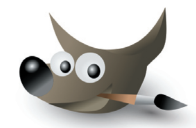
توزيعة جواثا العربية

تصدر تنقيحها الأول لإصدارة جواثا جنوم ١,٣ لكى تعرض فى معرض جايتكس دبي ٢٠٠٨ ضمن فعاليات صحارى الراعي الرسمي لتوزيعة جواثا لينُكس فى ١٩ من شهر أكتوبر الحالى ... ويارِيت ياسامر تضيف موقع للتوزيعة أنا معرفلهاش موقع بصراحة

مكتبة الإسكندرية ولينكس

كما كانت مكتبة الإسكندرية العريقة أحد أهم مصادر المعرفة فى العالم القديم فهى تطل علينا فى العصر الحديث وفى القرن الواحد والعشرين برعاية أنظمة لينُكس من خلال مجموعة من الندوات النصف شهرية يعدها وقدمها نخبة متميزة من خيرة الشباب العربى المهتمين بأنظمة لينكس وعلى رأسهم المهندس : محمد السيد
راجع نشرة المكتبة لآخر الندوات المقررة عن لينُكس فى شهر أكتوبر من هنا

http://www.bibalex.org/ARABIC/Calendar/ShowEvents.aspx?today=1







مقال بعنوان: الحقوق المرفوعة: المثالية الواقعية

تأليف : ريتشارد ستالمان
إعداد وترجمة: بدري دركوش

إن كل قرار يتخذه الإنسان ينبع من قيم هذا الإنسان وأهدافه؛ حيث يملك الناس العديد من الأهداف والقيم المختلفة، مثل: الشُهرة، والحُب، والبقاء، والمتعة، والحرية... هذه بعض الأهداف التي قد يمتلكها شخص جيّد، وعندما يكون الهدف هو مساعدة الآخرين، كما مساعدة الذات، وندعو ذلك مثاليّة.

إن عملي على البرمجيات الحرة يُحفّزه هدفٌ مثاليّ، هو: نشر الحرية والتعاون. أريد أن أشجِّع هذه البرمجيات حتى تنتشر وتستبدل البرمجيات المملوكة؛ والتي تمنع التعاون الذي سيجعل مجتمعنا أفضل.

رخصة "جنو" العموميّة، كما هي -أي كرخصةٍ مرفوعة Copylefted-، كل الشفرة المصدرية Source Code المضاف لبرنامج مُرخص بـ "جي بي إل" GPL يجب أن يكون برمجّية حرة، ولو كان في ملف منفصل. أنا أجعل كل شفراتي Codes متاحةً كبرمجيات حرة، وليست للاستخدام في برمجيات مملوكة، ولكي أشجع الآخرين على صناعة برمجيات حرة أيضًا.

لقد وجدت أنه إذا كان مطورو البرمجيات المملوكة يستخدمون حقوق الملكية Copyright ليمنوعنا من المشاركة؛ فإنّا -نحن المتعاونين- نستطيع استخدام حقوق الملكية لنُعطي المتعاونين الآخرين ميزتهم الخاصة: هم يستطيعون استخدام شفراتنا Codes بحرية.

ليس كل من يستخدم رخصة "جنو" العمومية لديه نفس الهدف. منذ عدة سنوات سُئل صديقٌ لي أن يُطلق برنامجًا ذا حقوقٍ مرفوعةٍ Copylefted تحت بنودٍ حرة غير ذلك، أي، حقوق غير مرفوعة Non-Copylefted؛ فأجاب بما يشبه الكلام التالي:
"أنا، أحيانًا أعمل على برمجيات حرة، وأحيانًا أخرى أعمل على برمجيات مملوكة، ولكن عندما أعمل على برمجيات مملوكة أنتظر أن أقبض راتبًا.

لقد كان ينوي أن يشارك عمله مع المجتمع الذي يتشارك بالبرمجيات، لكنه لم يجد أي سبب حتى يتبرع بالمنتجات الجالبة للأعمال، والتي تكون محددة لمجتمعنا. هدفه كان مختلفًا عن هدفي، لكنه قرر أن رخصة "جنو" العمومية مفيدةٌ لعمله أيضًا.

إذت أردت أن تُحقق شيئًا في هذا العالم فالمثالية لا تكفي. عليك اختيار طريقةٍ فعّالةٍ (عمليّة) للوصول إلى الهدف، بمعنًى آخر: أنت لا تحتاج إلا أن تكون "واقعيًا". هل "جي بي إلى" واقعيةٌ؟! لننظر إلى نتائجها.

انظر إلى GNU C++ Compiler. لماذا لدينا مصنف "سي بلس بلس" حر؟! فقط لأن رخصة "جنو" العمومية قالت إنه يجب أن يكون حرًا!. إن GNU C++ Compiler طُوّر من قبل تجمع صناعي -MMC- انطلاقًا من مصنف "جنو سي" GNU C Compiler.
إن MMC عادةً ما تجعل أعمالها مملوكةً بقدر ما تستطيع، لكنهم جعلوا الواجهة الأمامية Front End ل"سي بلس بلس" برمجية حرة، وذلك لأن رخصة "جنو" العمومية تنص على أنها الطريقة الوحيدة حتى ينشروا البرمجية.

إن الواجهة الأمامية ل"سي بلس بلس" تتضمن العديد من الملفات الجديدة، ولكن بما أنهم كانوا يتصلون بـ GCC، فيجب أن تنطبق عليهم رخصة "جي بي إل". إن الفائدة التي يحصدها مجتمعنا واضحة.

انظر إلى GNU Objective C. في البداية أرادت شركة NeXT أن تجعل هذه الواجهة الأمامية مملوكة. لقد عزموا على إطلاقها كملفات من نوع(o.) ، والتي تسمح للمستخدمين بربطهم ببيقة GCC، ظانين بذلك أنهم يلتفون حول (يتحايلون على) متطلبات رخصة "جي بي إل"، ولكن محامينا قال إن ذلك لن يعفيهم من المتطلبات؛ فذلك غير مسموح؛ لذلك جعلوا الواجهة الأمامية للمُهدِّف GNU Objective C برمجية حرة.

هذه الأمثلة حدثت منذ سنواتٍ عديدةٍ، ولكن رخصة "جِنو جي بي إل" تستمر في جلب المزيد من البرمجيات الحرة لمجتمعنا!.

هناك العديد من مكتبات "جِنو" محمية برخصة "جِنو" العمومية الصغرى GNU Lesser General Public License، ولكن ليس جميعها.

إحدى مكتبات "جِنو" المحمية برخصة "جنو جي بي إل" العادية هي Readline، والتي تحقق تحرير سطر الأوامر. في إحدى المرات، علمت عن برنامجٍ حرٍ صُمّم ليستخدم Readline، فأخبرت المطور بأن ذلك غير مسموح. كان يستطيع أن يزيل محرر سطر الأوامر من البرنامج، لكنَ ما فعله هو إصدار البرنامج تحت رخصة "جي بي إل"... إنه الآن برنامج حر!.

إن المبرمجين الذين يضيفون تحسينات إلى GCC، أو آيماكس، أو باش، أو لينوكس، أو أي برنامج مرخص برخصة "جي بي إل" يكونون عادةً موظفين من قبل شركاتٍ أو جامعات. عندما يريد المبرمج أن يرُدّ التحسينات إلى المجتمع، وأن يرى شفرته Code في الإصدار التالي؛ قد يقول رئيسه فيً العمل: "توقف مكانك... إن شفرتك ملكنا! نحن لا نريد أن نشاركه. لقد قررنا أن نحول نسختك المُحسّنة إلى منتج برمجيات مملوكة". هنا تأتي رخصة "جنو جي بي إل" للنجدة، يُري المبرمج لرئيسه في العمل أن هذا المنتج المملوك سوف ينتهك حقوق النشر Copyright. عند ذلك يصبح أمام رئيس العمل خياران فقط: إما أن يطلق الشفرة الجديدة كبرمجية حرة، أو لا يطلق شيئًا على الإطلاق. على الأرجح، دائمًا ما يُسمح للمبرمج أن يتابع ما كان يريد أن يفعله؛ وبالتالي يطلق الشفرة مع الإصدار التالي.

إن رخصة "جِنو جي بي إل" لا تُحابي أحدًا؛ فهي تقول "لا" لبعض الأشياء التي يرغب الناس في القيام بهَا. هناك بعض المستخدمين يقولون إن هذا شيءٌ سيءٌ. إن رخصة "جي بي إلى" تستثني بعض مطوري البرمجيات المملوكة، والذين يريدون أن ينضمُوا إلى مجتمع البرمجيات الحرة، ولكننا لا نستثنيهم من مجتمعنا، بل هم يختارون عدم دخوله. إن قرارهم بجعل البرمجيات مملوكة هو القرار بالبقاء خارج مجتمعنا. إن كونهم من مجتمعنا يعني الانضمام إلى التعاون معنا. نحن لا نستطيع أن "نجلبهم إلى مجتمعنا" إذا كانوا هم لا يريدون الانضمام إليه.

ما نستطيع أن نفعله هو أن نقدم لهم الحافز والدعوة للانضمام. إن رخصة "جِنو جي بي إل" مُصممة لتقدم دعوةً من قبل برمجياتنا الموجودة: "إذا جعلت برمجيتك حرة، تستطيع استخدام هذه الشفرة". بالطبع لن يربح كل شيءٍ، لكنه سيربح بعض الوقت الإضافي.

إن تطوير البرمجيات المملوكة لا يُساهم في مجتمعنا، ولكن مطوروه يرغبون عادةً بحسنة (تبرع) منّا. يستطيع مستخدمو البرمجيات الحرة أن يقدموا لمطوري البرمجيات الحرة ضروبًا من الغرور -والعرفان، والشكر، والتقدير-، لكن الأمر يكون مغريًا جدًا عندما يقول لك متاجر ما: "دعنا نضع تطبيقك في برنامجنا المملوك، وعند ذلك سوف يُستخدم برنامجك من قبل عدة آلافٍ من الناس!". قد يكون الإغراء كبيرًا، لكن، على المدى البعيد، نحن بحالٍ أفضل إن قاومنا هذا الإغراء.

عندما يكون الإغراء والضغط بشكلٍ غير مباشر يُصبح من الصعب تمييزه، وذلك من خلال منظمات البرمجيات الحرة، والتي تبنت سياسةً تسمح بتموين البرمجيات المملوكة. إنّ قضية اتحاد X أو X Consortium وخليفتها "المجموعة المفتوحة" The Open Group كمثالٍ، نشأت قبل الشركات التي تُصنّع البرمجيات المملوكة؛ فقد بذلوا جهدًا كبيرًا لسنواتٍ عدةٍ في حَثّ المبرمجين على عدم استخدام الحقوق المرفوعة. الآن قامت "المجموعة المفتوحة" بجعل X11R6.4 برمجيةً غير حرةٍ. إن من قاوم الضغط هو من أصبح سعيدًا بذلك الآن.

في سبتمبر ١٩٩٨، وبعد شهورٍ من إصدار X11R6.4 كبنود توزيع غير حرةٍ، قامت "المجموعة المفتوحة" بالتراجع عن قرارها، وقامت بإصدارها بنفس الرخصة الحرة (بدون حقوق مرفوعةٍ)؛ والتي استُخدمَت للإصدار السابق X11R6.3 . لكن هذا التراجع اللاحق لا يلغي الاستنتاجات التي وجدناها من الواقع الذي حصل بأن وضع قيود أمرٌ ممكن.

إن التفكير بشكل واقعي حول أهداف كبيرة وطويلة الأمد سوف يُقوّي عزيمتك على مقاومة الضغط. إذا ركزت تفكيرك على الحرية وعلى المجتمع الذي تستطيع بناءه إذا بقيت صامدًا، سوف تجد القوة للقيام بذلك.

"اصمد من أجل شيءٍ، أو تضيع من أجل لا شيء".

"Stand for something, or you will fall for nothing".

"وإذا قام معترضون راديكاليون وكارهون للحرية والمجتمع... إذا قال"أشخاصٌ واقعيون رغمًا عن أنوفهم" إن المنفعة هي المثالية الوحيدة... فقط، قم بتجاهلهم، واستخدم الحقوق المرفوعة دائمًا."

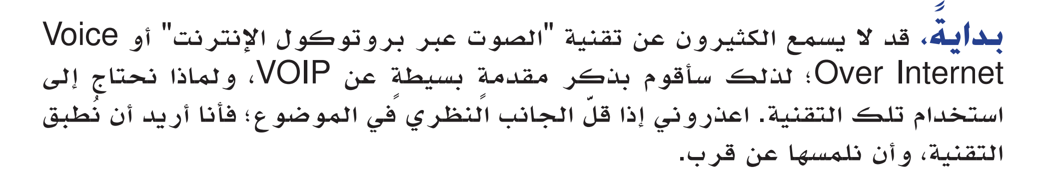
"And if cynics ridicule freedom, ridicule community...if "hard nosed realists" say that profit is the only ideal...just ignore them, and use copyleft all the same."

الهامش (من المترجم):
هناك تعريبان للمصطلح Copyleft: "الحقوق المتروكة" أو "الحقوق المرفوعة"، وقد اخترت المصطلح الثاني لأنه أقرب لمعنى المصطلح الإنكليزي، ولأن الأول يوحي بأنها مشاع، والله أعلم.



أنشئ خادم "فويب" مع OPENSER

للكاتب: محمد عقل



بدايةً، قد لا يسمع الكثيرون عن تقنية "الصوت عبر بروتوكول الإنترنت" أو Voice Over Internet؛ لذلك سأقوم بذكر مقدمة بسيطة عن VOIP، ولماذا نحتاج إلى استخدام تلك التقنية. اعذروني إذا قلّ الجانب النظري في الموضوع؛ فأنا أريد أن نُطبق التقنية، وأن نلمسها عن قرب.



الكل يعلم أن إيقاع الحياة يستم بالسرعة، يومًا بعد يوم؛ نتيجةً للنقلة الحضارية الهائلة في مجالات الاتصالات؛ سواءً كانت على صعيد الاتصالات الهاتفية، أو الاتصالات الخلوية، أو حتى الاتصالات عبر شبكات الإنترنت. نتيجةً لتلك النقلة، ظهرت العديد من التطبيقات والتقنيات العظيمة المَنْفَعة في مجال الاتصال عبر شبكات الإنترنت، وأهما، من وجهة نظري، "تقنية نقل الصوت عبر بروتوكول الإنترنت" أو ما يعرف ب VOIP.

"تقنية الفويب" هذه، هكذا تُنطق (Voyp)، تعني نقل المحادثات الصوتية باستخدام شبكة الإنترنت؛ عن طريق توجيه تلك المحادثات الصوتية؛ شأنها شأن الحزم العادية Packets، وبالتالي، لا يقتصر الأمر على شبكات الإنترنت Internet Networks، بل يمكن استخدام تلك التقنية داخل أي شبكة تستخدم بروتوكول الإنترنت، والتي تعرف ب IP-Based Networks، وبالتالي، يتم التعامل مع المحادثات الصوتية على أنها حزم يتم نقلها بإحدى بروتوكولات النقل، وأشهرها: TCP و UDP. حينها لا تستطيع طبقات الشبكة المختلفة Network Layers التفرقة بين حزم البيانات والحزم الصوتية.

لكن، قد يستغرب البعض، ويتسائل: ما الفائدة التي تعود علينا من استخدام تقنية VOIP؟!

للإجابة على هذا التساؤل، ومعذرةً، قد تكون الإجابة غير واضحة المعالم لمن ليست لهم درايةٌ بالاتصالات الهاتفية، تخيل معي السيناريو الافتراضي لشبكات الهاتف العامة، أو التي تعرف ب PSTN. حينما يريد مستخدم ما على تلك الشبكة الاتصال بمستخدم آخر، يحدث التالي:

في البداية حينما يريد المستخدم A أن يهاتف المستخدم B باستخدام PSTN، تقوم شبكة الهاتف أولاً بعمل إنشاء للاتصال قبل بدء المحادثة، ثم بعد ذلك تقوم الشبكة بحجز مسار محدد لكل من المستخدمين A و B لإجراء المكالمة؛ وبالتالي، كلما أراد المستخدم A محادثة المستخدم B، أو العكس، ستقوم شبكة الهاتف بحجز نفس المسار لكي تتم عملية الاتصال بشكلٍ سليم، وبذلك، دائمًا يتم حجز ذلك المسار بنفس السعة والموارد للمستخدم A حتى في حالة عدم احتياجه لذلك المسار في وقت ما، فمثلاً، قد يحتاج المستخدم A ذلك المسار بالنهار فقط، ولا يقوم باستخدامه بالليل، وبالتالي، يتم حجز الموارد الخاصة بالمستخدم A فقط، ولا يتم الاستفادة منها في حالة عدم استخدامها من قِبَله. شبكات الهاتف هذه تسمى علميًا ب Circuit-Switched Networks.

أما شبكات الإنترنت المُسماة ب Packet-Switched Networks، فالأمر مختلفٌ، بمعنى أن موارد الشبكة والمسارات المختلفة له تكون مشتركةً بين المستخدمين أو Shared Network Resources؛ ويعنى هذ أنه في حالة عدم وجود أيّ من المستخدمين لتلك الموارد يستطيع مستخدمٌ آخرٌ على نفس الشبكة استخدام تلك الموارد، في حالة عدم اسخدامها من قبَل مستخدِمين آخرين.

أضرب مثالاً عمليًا على ذلك:

نفترض أنه يوجد لديك حيزٌ تردديٌ Bandwidth، مثلاً، واحد ميجا، ولديك عشرة مستخدمين؛ فكيف يمكن تقسيم هذا الحيز الترددي على كلٍ من المستخدمين في شبكات الهاتف العادي PSTN وفي شبكات الإنترنت؟

أولًا: بالنسبة لشبكات الهاتف العادي PSTN. سنقوم بتقسيم ذلك الحيز الترددي، وهو ١ ميجا على العشرة مستخدمين، وبالتالي، يكون نصيب الفرد الواحد من مورد الشبكة هو ١٠٠ كليو، وتكون تلك ال ١٠٠ كيلو هي نصيب الفرد على الدوام، ولا يستطيع أن يأخذ أكثر أو أقل منها، واحتمال استخدام العشرة مستخدمين هؤلاء لمواردهم المحجوزة لهم في وقت واحد صغيرٌ جدًا، ولنفترض أن واحدًا من هؤلاء العشرة هو المستخدم الوحيد لموارده كاملةً، وباقي المستخدمين غير متواجدين!. معنى ذلك أن نسبة ٩٠٪ من موارد الشبكة محجوزةٌ بدون فائدةٍ، ولا يمكن استخدامها.

ثانيًا: شبكات الإنترنت، أو أي شبكة تستخدم بروتوكول الإنترنت، إذا كان الحيز الترددى ١ ميجا، يتم مشاركتها بين المستخدمين العشرة، وذلك أن احتمال دخول المستخدمين العشرة في نفس الوقت صغيرٌ جدًا، ويصل إلى نسبة ضئيلة أيضًا، ولنفترض أنه يوجد مستخدمٌ واحدٌ من العشرة موجودٌ Online؛ فسيكون بإمكانه التمتع بالحيز الترددي Bandwidth بالكاملً في حال عدم استخدامه من قِبل المستخدمين الآخرين، وذلك لأن شبكات الإنترنت تعمل بمدأ المورد عند الطلب أو Resources on Demand.

طبعًا المقارنة البسيطة التي قمت بسردها، في السطور السابقة، ما هي إلا مجرد نقطة في بحر من المقارنات التي تتم بين كل من مؤيدس شبكات الاتصال الهاتفية أو Circuit-Switched Networks وبين شبكات الإنترنت أو Packet-Switched Networks.

وبالتالى، نظرًا لأن "تقنية الفويب" تعمتد في طبيعة عملها على استخدام الشبكات التي تعمل ببروتوكول الإنترنت؛ فإنها تتمع بنفس ميزة المورد عند الطلب أو Resource on Demand، وهنا نستطيع استخدام الحيز الترددي بشكلٍ فعّالٍ وأكثر كفاءةً من شبكات الهاتف العادي PSTN.

كما تتميز "تقنية الفويب" أيضًا بخاصيةٍ مهمةٍ جدًا وهي رخص التكلفة!، نعم، تخيل كل ما تحتاجه مثلاً لعمل VOIP-Server لا يتعدى كونه وجود جهاز حاسب لديك مع بعض البرامج مفتوحة المصدر والمجانية، أي دون مقابل مالي. كذلك البنية التحتية للتقنية موجودةٌ فعلاً، ولا نحتاج إلى موارد إضافية.

أعلم أن تلك المقدمة بسيطةٌ جدًا في التعريف بمفاهيم كثيرة ذكرتها، لكن، اعذروني فعلاً؛ فكل مصطلح قد يحتاج إلى موضوع كامل، وسنخرج عن صلب الموضوع، فأرجو المعذرة منكم. ندخل الآن في صلب الموضوع: ماذا نحتاج لكي نتمكن من استخدام تلك التقنية؟

أولاً، بشكل أو بآخر قد تكون مُستخدمًا لتلك التقنية وأنت لا تشعر!. نعم؛ فبرامج المحادثة الصوتية والكتابية مثل: Skype، وYahoo Messenger، و MSN Messenger، وغيرها. هذه البرامج تعتمد في طريقة عمل المحادثات الصوتية الخاصة بها على تقنية VOIP.

لذلك، ما هو الجديد في موضوعنا؟!. سؤالٌ يستحق الإجابة فعلاً!.

الجديد في الموضوع أنك ستتمكن بنفسك من عمل خادم "فويب" VOIP-Server خاصٌ بك دون الحاجة إلى الآخرين، والفائدة التي ستعود عليك من ذلك أنه سيكون لديك عدة ميزات هائلة منها:

١. سيكون لديك مقسم هاتف داخلي مُعتمدٌ على بروتوكول الإنترنت أو ما يعرف ب IP PBX.
٢. تستطيع ربط "خادم الفويب" الخاص بك بشبكة الهاتف العادي PSTN .
٣. تستطيع عمل محادثات صوتٍ وصورةٍ بين المشتركين في الخادم الخاص بك.
٤. تستطيع عمل المحادثات الكتابية إلى جانب استخدام خدمة البريد الصوتى، خدمة الاجتماعات Conferences، خدمة الإعلانات Announcements، ....

تلك كانت بعض الميزات القليلة، والتي سوف تتمكن من الحصول عليها في حالة امتلاكك لخادم "فويب" خاص. قد يفرض سؤالٌ آخرٌ نفسه: ما هي المتطلبات اللازمة لعمل "خادم الفويب"؟

كل ما تحتاجه لتنفيذ الأمور التي نحن بصدد الحديث عنها في موضوعنا هو مجرد وجود جهاز حاسوب، وبعض البرامج التي سوف نقوم بتحميلها من الإنترنت، ويفضل أن يكون الحاسوب ذو إمكانياتٍ معقولةٍ؛ ليست بالكبيرة التي تكلفك المال الكثير، ولا بالقليلة التي قد تسبب لك شللاً نصفيًا نتيجة البطئ.

خطة العمل



لنناقش سويًا خطة العمل التي سوف نقوم بتنفيذها، والمطلوب منك إنجازه. سأقوم بسرد عدة نقاطٍ رئيسة:



أولاً: هل لديك شبكة حاسوب داخلية، وتود الربط بين تلك الأجهزة؛ لكي تستطيع عمل مكالمات داخلية (مثل سنترال داخلي)؟

ثانيًا: هل يوجد لديك نطاقٌ Domain خاصٌ بك، وتريد عمل خادم "فويب"؛ لكي تقوم بالربط بين مجموعة من المشتركين؟

ثالثًا: هل تنوي الربط مع شبكات الهاتف العادي PTSN؛ لكي تتمكن من الاتصال بالهواتف العادية، سواءً كانت داخل نطاق البلد، أو على نطاق عدة دول؟

في البداية، لكل خطة ذكرتها سابقًا الإعدادات الخاصة بها، ولكن سيكون هناك قاسمٌ مشتركٌ في الإعدادات بين تلك الخطط، فمثلاً: الخطة الأولى ستحتاجها إذا كنت تعمل في شركة تحتوى على عدد من الموظفين يعملون على حواسيب مرتبطة بشبكة؛ فسيكون باستطاعتك امتلاك مقسم هاتف داخلي مماثلٌ تمامًا للجهاز الذي يوجد داخل الشركة الخاصة بك، وسيمكنك الآن أن تخبرَ رئيسك في العمل أنكم، بعد اليوم، لن تحتاجوا إلى ذلك الجهاز (مكانه في ال Trash).

ملحوظة:

الشركات الكبيرة ليست بحاجة إلى تغيير السنترال الموجود لديها والبنية التحتية لهذا السنترال بوجود IP PBX. يمكنها الاحتفاظ حتى بالأرقام الداخلية للتحويلات، وباستخدام ال IP PBX سيتم تحويل المكالمات الداخلية من خلال الشبكة داخل الشركة، وتسمح للمستخدمين بمشاركة نفس التحويلات الداخلية القديمة بالشركة مع الخطوط الخارجية.

في الخطة الثانية: إذا كنت تمتلك نطاقًا Domain خاصًا بك، وتريد توسعة الأمر لربط عدة مستخدمين داخل عدة دولٍ مختلفة؛ فحينها ستحتاج إلى بعض الإعدادات الإضافية، وبعض الأمور التي ستقوم بوضعها في الحسبان.

أما أخيرًا، بالنسبة للخطة الثالثة، وهي: الربط مع شبكات الهاتف العادي، حينها ستحتاج إلى بعض العتاد الإضافي، وأعني بالعتاد الإضافي أن تمتلك PSTN Gateway؛ لكي تتمكن من الربط مع ال PSTN.

لذا، سيكون محور الحديث في موضوعنا عن كيفية تنفيذ الخطتين الأولى والثانية، واللتان قد يكون كثيرٌ منا يمتلك المقومات الخاصة بهما. أما الخطة الثالثة، فنظرًا لكونها تتطلب بعض التكلفة؛ لن نتطرق إليها.

خادمات الفويب (برمجيات حرة)

برامج "خادمات الفويب" كثيرة جدًا. منها ما هو تجاريّ، ومنها ماهو مجانيّ مفتوح المصدر. أشهر "خادمات الفويب" المجانية والمفتوحة المصدر هو Asterisk، يليها خادم اسمه OPENSER. إن شاء الله سيكون محور حديثنا هو OPENSER خلال السطور القادمة، وكما ذكرت، سيكون الكلام عمليًا أكثر منه نظريًا؛ نظرًا لأن الموضوع متشعب، وكل مصطلح يحتاج إلى فقرات لشرحه. نريد عمل "خادم الفويب" خطوةً بخطوةٍ؛ كي يتمكن الفرد العادي من إنشاء خادمٍ يخصه، سواءً داخل LAN، أو على مستوى النطاقات العامة Public Domains.

تثبيت OPENSER على توزيعة Debian GNU/Linux

بعض الأمور أفترضها:

توجد لديك توزيعة Debian GNU/Linux مثبتةٌ بالفعل؛ فعملية الشرح ستكون عليها.

معرفة كيفية التعامل مع الحزم: من إضافة، وتثبيت على التوزيعة.

أود الإشارة إلى أنه في حالة قيام أي شخص بتنفيذ الشرح على أية توزيعةٍ أخرى، وحدثت معه مشاكل فأنا غير مسئول عن ذلك؛ سواءً كانت هذه المشاكل هي عدم توافر حزمةً معينة، أو حدوث مشكلةٍ أثناء تنفيذ إحدى الخطوات في عملية ال Compiling ، ...إلخ.

السطور السابقة تناولنا فيها مقدمةً بسيطةً عن "الفويب" (VOIP)، وشرحنا في تلك المقدمة معنى كلمة VOIP، والفائدة التي قد تعود علينا من استخدام تلك التقنية، إلى آخره من تلك المفاهيم. في هذا الجزء، إن شاء الله، سنبدأ في تثبيت "خادم الفويب"، وستكون عملية التثبيت على توزيعة Debian، ومن أراد تثبيت OPENSER على أية توزيعة أخرى فسأذكر بعض النقاط الرئيسة، والتي تُمكّنه من تثبيت الخادم، بدون أية مشاكل، إن شاء الله.

متطلبات التثبيت (هذه تخص كل التوزيعات):

```
GCC
bison
flex
GNU make
sed and tr (used in the makefiles)
GNU tar
GNU install
libmysqlclient
openssl
```



ملحوظة: متطلبات التثبيت هذه تكفي فقط بوجود دعم لتوثيق المستخدمين على OPENSER باستخدام MySQL. توجد متطلبات أخرى إذا كنت تريد إضافة مهامٍ أكثر ليقوم بها OPENSER. جدير بالذكر أن أسماء الحزم السابقة هي أسماء الحزم بشكل عام.

أفترض أيضًا أنك قمت بتعديل ملف sources.list الخاص بك على التوزيعة، وإذا لم تكن قد فعلت؛ فإليك الطريقة:

بدايةً، قم بفتح مُحاكي الطرفية لديك Terminal. بعد ذلك قم بالدخول بحساب المستخدم الجذر root، ونفذ الأمر التالي:

```
nano /etc/apt/sources.list
```

بعد ذلك، قم بإضافة المصادر التالية إذا لم تكن موجودة عندك:

```
deb http://http.us.debian.org/debian etch main
deb-src http://http.us.debian.org/debian etch main
```

ملحوظة: بإمكانك استخدام أي مُحرر نصوصٍ آخر بدلاً عن nano. البعض يُفضل vi أو Vim. اختر ما يناسبك أو ما تستطيع استخدامه.

بعد ذلك، قم بتنفيذ الأمر التالي لتحديث ملف المصادر sources.list عن طريق الأمر:

```
apt-get update
```

الآن، سنقوم بتثبيت الحزم التي ذكرناها سابقًا، من خلال apt-get:

```
apt-get install gcc bison flex make openssl libmysqlclinet15-dev mysql-server
```

ملحوظة: قد توجد لديك بعض تلك الحزم السابقة، لكني أفترض أنها غير موجودة؛ وبالتالي إذا وُجدت الحزم فعلاً، وكانت هناك إصداراتٌ حديثةٌ منها؛ سيتم عمل ترقية لتلك الحزم، والتي لم تكن موجودة سيتم تثبيتها.

بعد التأكد من تثبيت الحزم السابقة على التوزيعة؛ سنقوم الآن بتحميل OPENSER من على الإنترنت، ثم نقوم بتثبيته، ونبدأ بخطوة التحميل أولاً:

نقوم بالولوج إلى المسار /usr/src، ثم نقوم بتنزيل الملف:

```
debian:~# c d /usr/src/; wget -c http://www.openser.org/pub/openser/1.2.2/
src/openser-1.2.2-tls_src.tar.gz
```

بعد ذلك، نقوم بفك الضغط عن الملف:

```
debian:/usr/src# tar -zxf openser-1.2.2-tls_src.tar.gz
```

ثم نقوم بحذف الملف المصدر لأنه لم تعد له قيمة (الخيار متروك لك):

```
debian:/usr/src# rm openser-1.2.2-tls_src.tar.gz
```

الآن، نقوم بالولوج إلى المجلد الذي يحتوي على الشفرة المصدرية ل OPENSER:

```
debian:/usr/src# cd openser-1.2.2-tls
```

بعد ذلك، سنقوم بعمل بعض التعديلات على ملف **Makefile**؛ لإضافة الmodule، الذي سوف يكون حلقة الوصل بين OPENSER، وبين محرك قواعد البيانات MySQL:

```
debian:/usr/src/openser-1.2.2-tls# nano Makefile
```

بعد ذلك، نقوم بالبحث عن السطر الذي توجد فيه كلمة mysql:

```
exclude_modules?=       jabber cpl-c mysql pa postgres osp unixodbc \
```

ثم نقوم بحذف كلمة mysql؛ ليصبح السطر بالشكل التالي:

```
exclude_modules?=       jabber cpl-c pa postgres osp unixodbc \
```

بعد ذلك، سنبدأ في تجميع الشفرة المصدرية الخاصة ب OPENSER، أو OPENSER Compiling:

```
debian:/usr/src/openser-1.2.2-tls# make prefix=/ all
```

ثم نقوم بتثبيت الملفات التي تم عمل لها عملية التجميع من الشفرة المصدرية:

```
debian:/usr/src/openser-1.2.2-tls# make prefix=/ install
```

بعد ذلك، نقوم بإنشاء مجلدٍ جديدٍ، على المسار /var/run؛ لكي نضبط المسار الخاص بملف رقم العملية ل OPENSER:

```
debian:~# mkdir /var/run/openser
```

وهكذا، انتهت عملية التثبيت بنجاح، وأصبح لديك "خادم الفويب" OPENSER على التوزيعة. الخطوة القادمة هي وضع OPENSER كعملية يتم تشغيلها أثناء إقلاع Debian GNU/Linux، ونبدأ على الفور بتنفيذ التالي من محاكي الطرفية:

```
debian:~# cd /usr/src/openser-1.2.2-tls/packaging/debian

debian:/usr/src/openser-1.2.2-tls/packaging/debian# cp openser.default /etc/
default/openser

debian:/usr/src/openser-1.2.2-tls/packaging/debian# cp openser.init /etc/
init.d/openser; cd

debian:~# update-rc.d openser defaults 99
```

بعد ذلك، قم بالتعديل على ملف openser.cfg، والذي يمثل ملف الإعدادات الرئيس، ونقوم بالبحث عن السطر #fork=no، ثم نحذفه، وبعد ذلك نقوم بمنح الملف openser على المسار /etc/init.d التصاريح اللازمة للتنفيذ:

```
debian:~# chmod 755 /etc/init.d/openser
```

بعد ذلك، نقوم بالتعديل على ذلك الملف لتعديل المسار الذي يوجد به الملف الرئيس للخادم:

```
debian:~# nano /etc/init.d/openser
```

نقوم بالبحث عن السطر:

```
DAEMON=/usr/sbin/openser
```

ثم نجعله كالتالي:

```
DAEMON=/sbin/openser
```

أخيرًا، نقوم بالتعديل على الملف openser الموجود على المسار /etc/default؛ لتعديل اسم المستخدم والمجموعة اللذان سيكون لهما الحق في تشغيل الخادم أثناء عملية الإقلاع، مع ضبط بعض الخيارات الأخرى:

```
debian:~# nano /etc/default/openser
```

ونقوم بتعديل القيم التالية لتصبح بالشكل التالي:

```
RUN_OPENSER=yes
MEMORY=128
USER=the_user_you_want
GROUP=the_group_you_want
```

بعد أن ننتهي من ضبط تلك الإعدادات؛ نحفظ التغييرات التي أجريناها، ثم نقوم بعمل إعادة تشغيل للتوزيعة؛ لنتأكد من أن كل شيء أصبح على ما يرام، وأنه لا توجد هناك أية مشكلة، وبعد بدء التشغيل مرةً أخرى، تستطيع التأكّد من أن OPENSER يعمل فعليًا باستخدام الأمر ps بالشكل التالي:

```
debian:~# ps aux | grep openser
```

الآن، سنتطرق إلى كيفية ضبط إعدادات الخادم، وكيفية التعديل على ملف الإعدادات الرئيس. لكن، في البداية، سوف أتطرق إلى شرح بعض المفاهيم التي قد تلتبس لدى البعض:

أولاً، OPENSER واعتماده على مبدأ: "واحدٌ للكل، والكل لواحدٍ!".

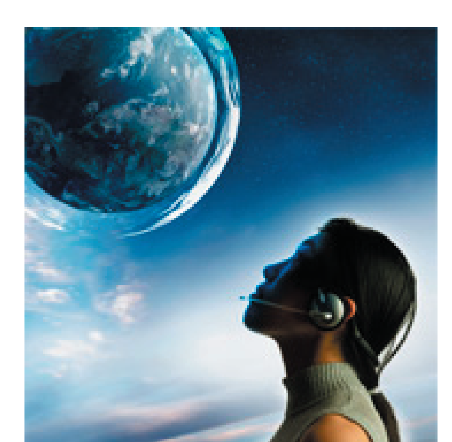
أعتقد أن البعض قد بدأ يفكر: هل جننت؟!، لا ليس بعد، ما أقصده من العبارة السابقة نقطة مهمة تخص "خادم الفويب" OPENSER، وهي كيف تم إنشاء OPENSER ليعمل خادمًا للفويب، بمعنى أن OPENSER في حد ذاته كحزمة أو برنامج لا تتعدى مساحته، كشفرات برمجية، حاجز ال ٥٠٠ كيلو بايت!. معقول؟!. لكن هل ذلك البرنامج، أو الحزمة، تستطيع عمل كل الوظائف التي ذكرتها سابقًا في بداية موضوعنا؟. قطعًا لا!.

"خادم الفويب" OPENSER، كحزمة، لا يستطيع فعل أي شيء على الإطلاق، لكن الفكرة العبقرية هنا، والتي هي جزءٌ من سبب انتشاره؛ هي اعتماده على نماذج خارجية External Modules تقوم بالربط بين OPENSER والبرامج المختلفة، فعلى سبيل المثال: نريد أن نستخدم OPENSER مع نظام إدارة قواعد البيانات MySQL؛ هنا يأتى دور النموذج mysql.so، والذي تكون مهمته الرئيسة عمل الاتصالات اللازمة مع MySQL، وهكذا، قِس الأمر على نفس هذا المنوال حينما تريد مثلاً أن تقوم بربط برنامج خارجي مع OPENSER؛ تقوم ببناء Module يقوم بدورِ الوسيط بين ذلك البرنامج وبين OPENSER.
ملحوظة: بعض ال External Modules تكون معتمدة على بعضها البعض بشكل أو بآخر؛ فتنبه لتلك النقطة لأننا سوف نحتاج إليها لاحقًا.

إذًا، السؤال الذي يفرض نفسه علينا الآن: كيف نقوم، مثلاً، بإضافة نموذج خارجي جديد New External Module لخادم الفويب OPENSER، أو قد نطرح السؤال بشكل آخر: هل يوجد ل OPENSER ملَفٌ معينٌ يتم التعديل من خلاله على خصائص OPENSER؛ لإضافة أو حذف External Module معين؟

نعم، لدى OPENSER ملف ضبط الإعدادات الخاصة، ويدعى openser.cfg، وهذا الملف هو الملف الرئيس، أو العقل المدبر ل OPENSER، وبدونه OPENSER لا يساوى شيئًا. فمثلاً عند تشغيل OPENSER يقوم أولا بفحص ملف الإعدادات openser.cfg؛ ليتأكد من صحة وضع الإعدادات بشكل سليم، كما يتأكد من النماذج الخارجية المُضافة ليقوم بتحميلها أم لا، وبالتالي نستطيع القول بأن OPENSER بدون openser.cfg يكونُ عديم القيمة، ولا فائدة منه.

المرحلة الأولى من تشغيل OPENSER



الملف openser.cfg يتم إنشاؤه بشكل إفتراضيّ بعد الانتهاء من تثبيت OPENSER، وقبل الشروع في التعديل على الملف، سنقوم ببعض الخطوات الإضافية لكي نستعمل OPENSER دون أية مشاكل.
نبدأ أولاً بتعريف SIP_DOMAIN للجهاز الذي سوف يتم تثبيت OPENSER عليه، كما يلي:

```
debian:~# export SIP_DOMAIN='localhost'
```

ملحوظة: في كل مرة سوف نحتاج إلى عمل export لـ SIP_DOMAIN في حال إعادة تشغيل الجهاز؛ ولذا سوف نقوم بإضافة الأمر export داخل ملف bashrc، والذي يكون داخل مجلد ال home directory باسم المستخدم الخاص بك، وتستطيع الوصول إليه بالشكل التالى:

```
debian:~# nano ~/.bashrc
```

بعد ذلك، قم بإضافة هذا السطر في آخر الملف:

```
export SIP_DOMAIN='localhost or mydomain.domain.com'
```

أو، تستطيع عمل ذلك مباشرةً باستخدام echo بالشكل التالى:

```
echo "export SIP_DOMAIN='localhost'
" >> ~/.bashrc
```

ثم قم بحفظ الملف، وبعدها لن نحتاج إلى عمل export لـ SIP_DOMAIN مرةً أخرى.

ملحوظة: يمكنك وضع النطاق الخاص بك مكان كلمة localho إذا كنت تمتلك نطاقًا؛ وذلك لربط عدة مستخدمين على مستوى الشبكة واسعة النطاق (WAN).

المرحلة الثانية: OPENSER with MySQL Support

الآن، سنبدأ في تجهيز MySQL للعمل مع OPENSER، وذلك بإنشاء قاعدة بيانات خاصةٍ بـ OPENSER، وسيكون ذلك من خلال السكربت openser_mysql.sh. نقوم على الفور بتنفيذ الأمر التالي في الطرفية:

أولاً: سنقوم بوضع كلمة مرور للمستخدم الجذر root؛ لتوفير بعض النواحي الأمنية، وعدم العبث بمحرك قاعدة البيانات من قبل أي شخصٍ آخر:

```
debian:~# mysqladmin -u root password "any_pass"
```

ثانيًا: نقوم بإنشاء قاعدة البيانات الخاصة بـ OPENSER من خلال تنفيذ الملف openser_mysql.sh على المسار /sbin بالشكل التالي:

```
debian:~# cd /sbin
debian:/sbin# openser_mysql.sh create
```

سيظهر لك في محاكي الطرفية أو شاشة الكونسول التالي:

```
MySQL password for root:
creating database openser...
Core OpenSER tables succesfully created.
Install presence related tables ?(y/n):y
creating presence tables into openser...
Presence tables succesfully created.
Install extra tables - imc,cpl,siptrace,domainpolicy ?(y/n):y
creating extra tables into openser...
Extra tables succesfully created.
Install SERWEB related tables ?(y/n):y
Domain (realm) for the default user 'admin': localhost

creating serweb tables into openser...
SERWEB tables succesfully created.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                                 !
!                   WARNING                       !
!                                                 !
! There was a default admin user created:         !
!    username: admin@localhost
!    password: openserrw
!                                                 !
! Please change this password or remove this user!
! from the subscriber and admin_privileges table.!
!                                                 !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

ثالثًا: سوف نبدأ سويًا في معرفة كيفية ضبط إعدادات OPENSER عن طريق الملف openser.cfg، والذي يكون موجودًا داخل المسار التالي:

```
/etc/openser/openser.cfg
```

لتوفير بعض الوقت والجهد، قمت بضبط ملف إعدادات كامل يتضمن إضافة دعم قواعد البيانات والتوثيق من خلالها باستخدام محرك قواعد البيانات MySQL، وتستطيع تحميل الملف من خلال الرابط:

```
debian:~# cd /etc/openser; wget http://muhammad.akl.googlepages.com/auth-mysql.cfg
```

بعد ذلك، قم بنسخ الملف auth-mysql.cfg ليعمل بدلاً عن الملف الحالي openser.cfg:

```
debian:/etc/openser# cp auth-mysql.cfg openser.cfg
```

الآن، سنقوم بالتعديل على الملف الجديد openser.cfg، ونقوم بالبحث عن بعض السطور:

```
debian:/etc/openser# nano openser.cfg
```

نبحث عن الجملة التالية:

listen= رقم ال IP الخاص بالخادم لديك، أو جهازك الشخصي.

كذلك الجمل التالية في نفس الملف:

```
if (!www_authorize("", "subscriber")) {
              www_challenge("", "0");
```

ونضع بين علامتي التنصيص "" اسم النطاق الخاص بنا، سواءً كان على مستوى النطاق المحلي **localhost** أو النطاق العام public domain، وبعد التعديل يكون الشكل النهائي:

```
if (!www_authorize("localhost or yourdomain.domain.com", "subscriber")) {
              www_challenge("localhost or yourdomain.domain.com", "0");
```

لكن، ينبغي التنبيه على أن القيم المدخلة الخاصة بالنطاق لابد وأن تكون متطابقةً مع القيمة التي قمنا بإعطائها للمتغير SIP_DOMAIN سابقًا؛ فلتنتبه لذلك.

رابعًا: الخطوة الرابعة والأخيرة هي ضبط إعدادات الملف openserctlrc على المسار /etc/openser عن طريق تحريره:

```
debian:/etc/openser# nano openserctlrc
```

توجد بعض المتغيرات الأخرى التي سوف نقوم بتعديلها أيضًا، وتوفيرًا للوقت والجهد مرةً أخرى؛ قمت بتوفير ملفٍ مُعدّلٍ جاهزٍ، وتستطيع تحمليه من على الرابط:

```
debian:/etc/openser# wget http://muhammad.akl.googlepages.com/openserctlrc
```

بعد تحميل الملف، سيكون اسمه openserctlrc.1 . قم بنسخه مكان الملف الأصلي:

```
debian:/etc/openser# cp openserctlrc.1 openserctlrc
```

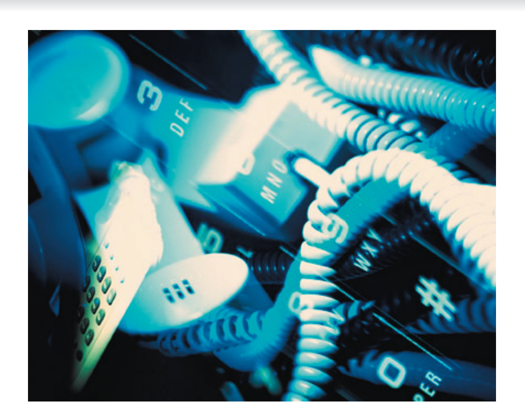
مع ملاحظة أنه يوجد داخل الملف متغير اسمه SIP_DOMAIN، تأكد من ضبط قيمته أيضًا لكي تتوافق مع القيم التي أدخلناها سابقًا

عمل اختبار للخادم

إلى هنا، وصل قطارنا إلى المرحلة الأخيرة؛ وهي عمل اختبار للخادم الخاص بنا، ومدى كفاءة العمل، وسنختم هذه المرحلة بإضافة مستخدمين إلى قاعدة البيانات الخاصة بنا؛ لكي نضمن عدم العبث بالخادم من قبل أشخاصٍ غير موثقين، وغير مسموح لهم باستعمال الخادم للمكالمات، وسيكون ذلك من خلال الأمر openserctl على الشكل التالي:

```
debian:~# openserctl add 1000 1000 1000@localhost
```

ونقوم بتكرار الجملة السابقة عدة مرات بحسب عدد المستخدمين الذين نريد السماح لهم باستخدام الخادم مع ملاحظة تغيير الرقم ١٠٠٠ بأرقام أخرى، مثلاً: متسلسلة، مثل: ١٠٠١ و ١٠٠٢.... وهكذا.



ملحوظة: بعد تنفيذ الأمر السابق سيتم طلب إدخال كلمة مرورٍ منك، وهي .openserrw

بعد ذلك، سنقوم باستخدام برامج هاتفية للتأكد من أن كل شيءٍ أصبح على ما يرام، ومن أمثلة تلك البرامج: برنامج ekiga، والذي يأتي بشكلٍ افتراضيّ مع كلٍّ من توزيعتي Debian GNU/linux و Ubuntu، وتستطيع الوصول إليه عن طريّق:

Application => Internet => Ekiga Softphone

بعد ذلك، ستظهر لك شاشة مساعد ضبط الإعدادات، أهملها وقم بالضغط على زر Cancel. بعد ذلك ستجد قائمةEdit بالأعلى، اختر منها Accounts. بعد ذلك ستظهر لك نافذةٌ أقصى اليمين، فيها كلمة Add، اضغط عليها؛ لإضافة مستخدمٍ جديد، ثم قم بملء البيانات التالية:

Account Name = الاسم الذي سوف يظهر للمستخدمين الآخرين

Protocol = SIP

Registrar = رقم الآيبي الخاص بالخادم أو الجهاز الشخصي

User = رقم المستخدم، والذي كمثال عليه ١٠٠٠

Password = كلمة المرور للمستخدم والتي قمنا بإدخالها، أيضًا ١٠٠٠

Authentication login = هنا سوف تكون القيمة نفس قيمة كلمة المرور، وفي المثال ١٠٠٠ أيضًا

بعد إدخال البيانات بشكلٍ صحيح؛ ستظهر رسالة أسفل الشاشة الرئيسة للبرنامج؛ تفيد أن ذلك المستخدم مسجل لدى الخادم الذي رقم ال IP الخاص به xxx.xxx.xxx.xxx

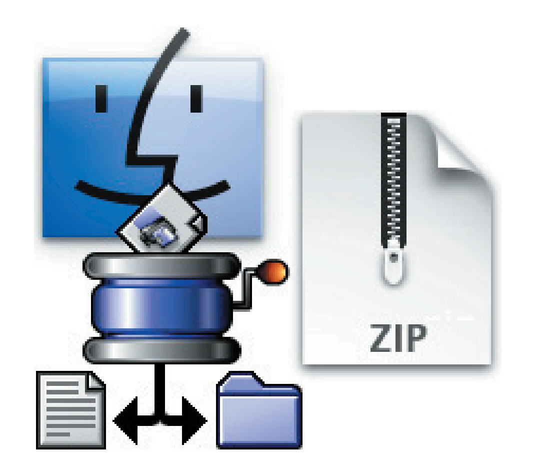
الآن، كل ما علينا فعله هو الاتصال بأحد المستخدمين المسجلين لدينا على نفس الخادم، وهنيئًا لك، امتلك "خادم فويب" كامل!.

أتمنى أن يكون الموضوع قد حاز رضاكم، ولأي استفسار بخصوص الموضوع؛ رجاءً، المراسلة على: m.akl@linuxac.org أو من خلال مواضيع النقاش في منتديات مجتمع لينوكس العربي.



التعامل مع الأراشيف والملفات المضغوطة في جنو / لينوكس

للكاتبة: حنان العربي



لا يخفى على الكثيرين من مستخدمي الحاسوب مدى أهمية استخدام طرق الأرشفة وضغط الملفات المختلفة في نظام التشغيل الخاص بنا ، لما يوفره ذلك من سهولة تصنيف للملفات والمجلدات وتقليل المساحة المستخدمة في الأقراص الصلبة أو وحدات التخزين الاخرى.
ولأهمية هذا الموضوع لمستخدمي أنظمة تشغيل جنو / لينوكس فقد ارتأيت طرح العديد من الأسئلة والاجوبة التي قد تتبادر إلى ذهن الكثير وخاصة المنتقلين الجدد لعالم الحرية باستخدامهم لنظام تشغيل جنو / لينوكس حول هذا الأمر.

أولاً. باستخدام سطر الأوامر

سؤال: في نظام التشغيل "ويندوز" كنا نستخدم برنامج WinZip لضغط وفك الضغط عن الملفات، ماذا عن "لينوكس"؟ كيف أستطيع ضغط وفك ضغط الملفات ذات الامتداد .zip في "لينوكس"؟
علمًا بأنني أستخدم توزيعة مبنية على "ديبيان" Debian.

الإجابة: لدينا في "لينوكس" برنامجان: أحدهما لضغط الملفات، والآخر لفك الضغط عنها. لكنها لا تأتي تلقائيًا مع التوزيعة. يمكن تنصيبها من خلال سطر الأوامر.
باستخدام صلاحيات المستخدم الجذر root، نفذ الأمرين التاليين:

```
# apt-get install zip
# apt-get install unzip
```

وإذا كنت تستخدم توزيعة مبنية على "ريد هات" Red Hat، مثل: "فيدورا" Fedora، يمكنك القيام بذلك باستخدام الأمر yum:

```
# yum install zip
# yum install unzip
```

الأمر zip:
أداة لضغط أو أرشفة الملفات.

الأمر unzip:
أداة لترتيب أو اختبار أو استخراج الملفات من أرشيف بالامتداد .zip.

بعض الأمثلة والتطبيقات على استخدام هذين الأمرين:

- ضغط الملفات، باستخدام الأمر zip:
أ- عمل الأرشيف data.zip، وضع جميع الملفات التي في المسار فيه.

```
$ zip data *
```

ملاحظة: لست بحاجة إلى إضافة الامتداد .zip أو أي لاحقة للأمر السابق، لأنها تضاف بشكل تلقائي مع هذا الأمر.
ب- ضغط مسار بأكمله (ويشمل ذلك جميع المسارات الفرعية).

```
$  zip -r data *
```

- فك الضغط عن الملفات (استخراج الملفات)، باستخدام الأمر unzip:

أ- استخراج جميع الملفات التي في الأرشيف pics.zip، في نفس مسار الأرشيف.

```
$ unzip  pics.zip
```

ب- اختبار الأرشيف pics.zip؛ للتأكد من سلامته.

```
$ unzip -tq pics.zip
```

ت- استخراج الملف cv.doc من الأرشيف pics.zip.

```
$ unzip pics.zip  cv.doc
```

ث- استخراج جميع الملفات إلى المسار /tmp.

```
$ unzip pics.zip  -d /tmp
```

ج- ترتيب جميع الملفات التي في الأرشيف pics.zip على شكل قائمة.

```
$ unzip -l pics.zip
```

- استخراج الملفات المضغوطة من أكثر من أرشيف، في وقت واحد:
عندما نريد استخراج الملفات باستخدام Wild Card؛ يكون لدينا خيَاران:-

الأول: استخراج الملفات المضغوط من عدة أراشيف؛ باستخدام علامة التنصيص Short Version:

```
$ unzip '*.zip'
```

ملاحظة: كلمة *.zip توضع بين علامتي التنصيص؛ وذلك لتفادي تعرف سطر الأوامر عليها على أنها wild card characte.

الثاني: استخراج الملفات المضغوطة من عدة أراشيف باستخدام **Shell for loop (long version)** :

```
$ for z in *.zip; do unzip $z; done
```



- استخراج الملفات في مسار محدد، في نظامي Linux أو Unix:

سؤال: لدي الأمر unzip، ولدي الأرشيف package.zip، وأستطيع أن أستخرج الملفات باستخدام الأمر unzip package.zip، لكن الملفات تخرج في نفس المسار الموجود به الملف الأصلي (يقوم الأمر بعمل مجلد اسمه package، في نفس مسار الملف الأصلي، ويضع به جميع الملفات التي تم استخراجها من الأرشيف). كيف أستطيع استخراج جميع الملفات في مسار أقوم أنا بتحديده، مثلاً: /opt؛

الإجابة: أمر unzip يقوم بترتيب، واختبار، واستخراج الملفات من أرشيف يحمل الامتداد zip، وهو شائع في الأنظمة المبنية على MS-DOS.

وبشكلٍ تلقائيّ؛ يتم استخراج جميع الملفات في نفس مسار الأرشيف الأصلي، ولجعل الملفات تُستخرج في المسار الذي تقوم أنت بتحديده؛ يتوجّب عليك استخدام الأمر -d

نفترض دائمًا أنك لديك الإذن بالكتابة في هذا المسار، وتكون كتابة الأمر كالتالي:

```
unzip {.zip-file-name}-d {/path/to/extract}
```

مثال:

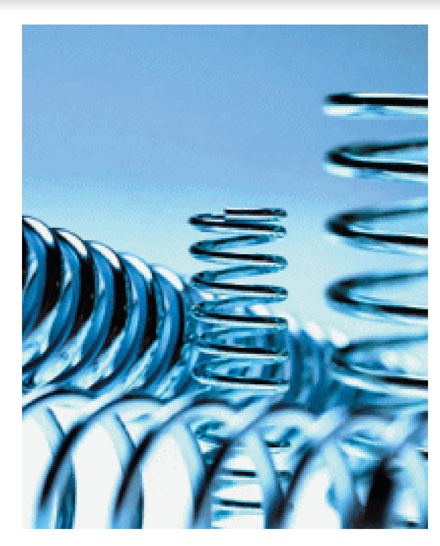
استخراج الملفات من الأرشيف package.zip إلى المسار /opt:

```
# unzip package.zip -d /opt
# cd /opt
# ls
```

إذا أردت إعادة تسمية مسار الملف؛ تنفذ الأمر التالي:

```
# mv package newname
```

التعامل مع الملفات التي تحمل الامتدادين gz و tar



سؤال: قمت بتحميل بعض الملفات من الإنترنت، وكل ملف كان يحمل الامتداد gz،كيف أستطيع فتح هذه الملفات باستخدام لينوكس؟

الإجابة: سوف تحتاج إلى برنامجgunzip / gzip، وهو يستخدم لضغط وفك الملفات التي بهذا الامتداد.
فك الضغط عن الملفات التي تحمل الإمتدادgz:

```
$ gunzip file.gz
```

أو

```
$ gzip -d  file.gz
```

ويمكن لمستخدمي GNU/Linux استخدام الخيار z مباشرةً كالآتي:

```
$ tar xvzf file.tar.gz
$ tar xvzf file.tgz
```

سؤال: كيف أستطيع ضغط مسارٍ كاملٍ في نظام UNIX/Linux باستخدام سطر الأوامر؟

الإجابة: من السهل جدًا القيام بذلك في نظام UNIX/Linux، وهذا مفيدٌ في عمل نسخة احتياطية للملفات، أو إرسال جميع الملفات في المسار دفعةً واحدةً عبر البريد الإلكتروني، أو حتى إرسال البرامج للأصدقاء. من الناحية التقنية يُسمى الملف الناتج بالأرشيف المضغوط.
إن الأمر tar هو الأفضل للقيام بتلك العملية، ويمكن استخدامه في خادمات UNIX/Linux البعيدة، وهو يقوم بأمرين:
١. صنع الأرشيف
٢. ضغط الأرشيف
ونحتاج للقيام بهذه العملية أمرًا مكتوبًا بهذه الصيغة:

```
tar -zcvf ArchiveName.tar.gz DirectoryName
```

حيث:
-z:
أرشيف مضغوط باستخدام برنامج gzip
-c: صنع أرشيف
-v: رؤية التفاصيل أثناء العملية
-f: اسم الأرشيف

مثال: لدينا مسار:

```
/home/jerry/prog
```

لضغط جميع الملفات التي في هذا المسار؛ ننفذ الأمر:

```
$ tar -zcvf prog-1-jan-2005.tar.gz /home/jerry/prog
```

إن هذا الأمر سيقوم بصنع أرشيف يحمل الاسم prog-1-jan-2005.tar.gz في المسار الحالي، وإذا أردت عكس العملية وفك الضغط عن الأرشيف ما عليك سوى كتابة الأمر التالي (الذي سوف يقوم باستخراج جميع الملفات من الأرشيف في نفس المسار):

```
$ tar -zxvf prog-1-jan-2005.tar.gz
```

بحيث:
-x: فك الضغط عن الأرشيف
وإذا أردت فك الضغط واستخراج جميع الملفات في مسارٍ محددٍ، مثلاً

```
/tmp
```

فإننا نحتاج إلى استخدام الأمر التالي:

```
$ tar -zxvf prog-1-jan-2005.tar.gz -C /tmp
$ cd /tmp
$ ls -
```

سؤال: كيف أستطيع استخراج ملفٍ أو مسارٍ من أرشيف من النوع tarball باستخدام سطر الأوامر في Linux أو UNIX؟
وكيف أستطيع استعادة ملفٍ واحدٍ من المسار /dev/st0 والذي يكون بالعادة خاص بالشريط Tape ؟

الإجابة: إن الأمر tar هو الذي يقوم بهذه العملية للأراشيف التي من هذا النوع،ويكون ذلك بكتابة الآتي، في سطر الأوامر:

```
tar xvf /dev/st0 filename
tar xvf /dev/st0 directory-name
tar xvf mytar.ball.tar filename
tar -zxvf mytar.ball.tar.gz directory-name
```

ولتمديد الملفات إلى المسار

```
tmp/
```

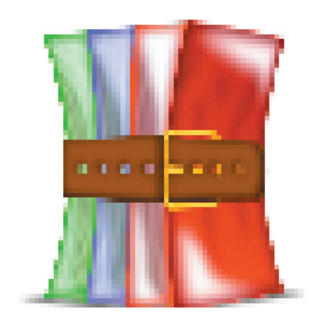
نكتب:

```
tar -zxvf mytar.ball.tar.gz -C /tmp filename
tar -zxvf mytar.ball.tar.gz -C /tmp dir-name
```

لقراءة ومعرفة المزيد عن كيفية التعامل مع أرشيف من هذا النوع نكتب:

```
man tar
```

كيفية التعامل مع الملفات التي تحمل الإمتداد rar



سؤال: كيف أستطيع فتح ملف يحمل الامتداد rar في أنظمة UNIX أو Linux؟
الإجابة: الملفات التي تأتي بهذا الامتداد هي عبارة عن أرشيفات مضغوطة؛ فإذا قمت بتحميل ملفات بهذا الامتداد من الإنترنت؛ فأنت بحاجة إلى برنامج ليفك الضغط عن هذه الملفات عن طريق استخدام الأمر unrar، وهذا البرنامج لايأتي تلقائيًا مع النظام، بل يجب تنصيبه عن طريق الأمر:

```
# apt-get install unrar
```

وهذا الأمر يكون لتوزيعة Debian، والتوزيعات المبنية عليها.

أما بالنسبة لتوزيعة RedHat، والتوزيعات المبنية عليها نستخدم الأمر:

```
# yum install unrar
```

أما إذا كنت تستخدم أي نظامٍ آخر مفتوح المصدر؛ يمكنك استخدام الأمر التالي:

```
# pkg_add -v -r unrar
```

وإذا لم تنجح أي من الطرق السابقة في تنصيب البرنامج؛ يمكنك تنزيل الحزمة الثنائية Binary Package من موقع rarlab الرسمي باستخدام الأمر:

```
$ cd /tmp
$ wget http://www.rarlab.com/rar/rarlinux-3.6.0.tar.gz
```

ثم تقوم بفك الضغط عن الحزمة باستخدام الأمر:

```
$ tar -zxvf rarlinux-3.6.0.tar.gz
```

سترى أن كلا الأمرين موجودين في المسار الفرعي لـrar فقط اذهب إلى هذا المسار باستخدام:

```
$ cd rar
$ ./unrar
```

ثم انسخ هذين الأمرين إلى المسار:/bin
باستخدام:

```
# cp rar unrar /bin
```

أمثلة وتطبيقات على استخدام الأمر unrar:

١.فك الضغط عن ملف بالامتداد rar في نفس مسار الملف الأصلي:

```
$ unrar file.rar
```

٢.إدخال الملف (l) داخل أرشيف يحمل الامتداد rar:

```
$ unrar l file.rar
```

٣.فك الضغط عن الملفات (x):

```
$ unrar x file.rar
```

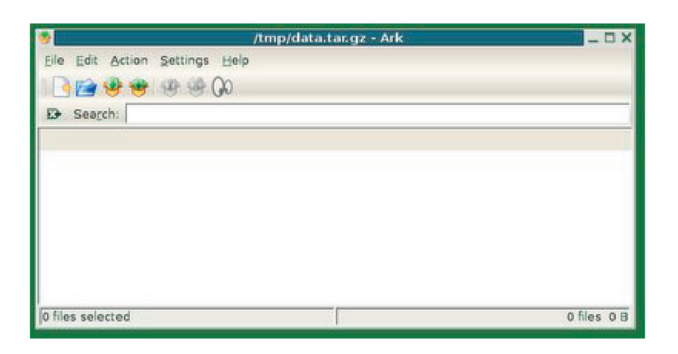
٤.اختبار (t) لصلاحية الأرشيف:

```
$ unrar t file.rar
```

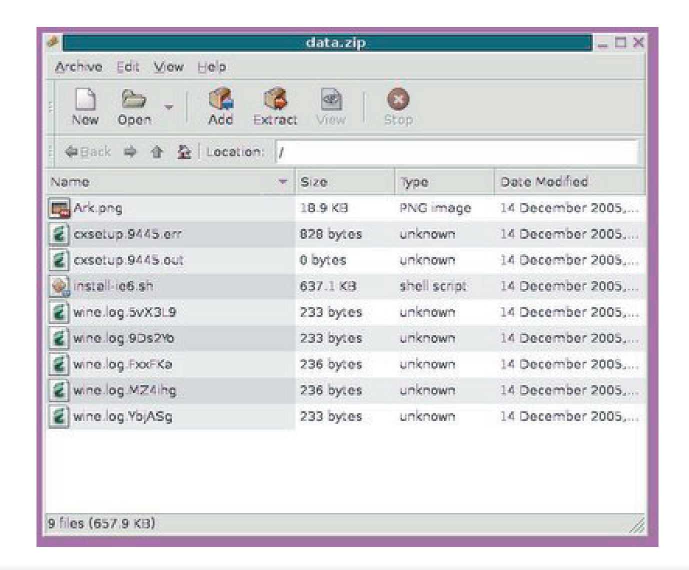
ثانيًا. باستخدام الواجهة الرسومية

يمكنك استخدام الواجهة الرسومية لضغط واستخراج الملفات المضغوطة.

١.الواجهة KDE:
برنامج ArK هو المسؤول عن التعامل مع و إدارة الملفات المضغوطة، ويمكن الوصول إليه عن طريق:
Application > Accessories
أو
البرامج > الملحقات

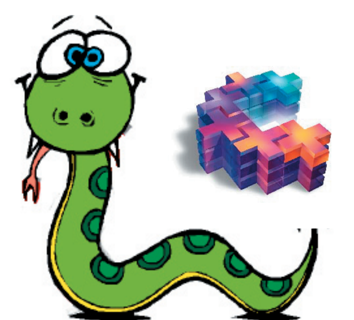


٢.الواجهةGNOME:
برنامج File Roller هنا هو المسؤول عن التعامل مع وإدارة الملفات المضغوطة.



"سايثون" تجمع بين "سي" و"بايثون"

للكاتب: مؤيد السعدي



في هذا المقال سنتحدث عن طريقة الجمع بين لغة "سي" ومكتباتها الكثيرة من جهة، وسهولة "بايثون" من جهة أخرى.

البعض يفضل مكتبة SWIG (اختصار. Simplified Wrapper and Interface انظر موقع www. swig.org)

وهي مكتبة تستطيع الربط بين "سي" والكثير من اللغات مثل:

Perl, PHP, Python, Tcl and Ruby

C#, Common Lisp (CLISP, Allegro CL, CFFI, UFFI), Java, Lua, Modula-3, OCAML, Octave and R

لكنني سأتحدث عن "سايثون" Cython (من الموقع www.cython.org) وهي بعكس SWIG تولد كود "سي" مستقل عنها يمكن عمل تصنيف له Compile بشكل طَبْعيّ. باختصار، "سايثون" Cython هي لغةٌ مُستَلهمةٌ من لغة "بايثون"، تصلح فيها كلُ تعابير "بايثون" (تقريبًا). ولكنها، بعكس "بايثون"ً، تحتوي على أنواع بيانات "سي" الطَبْعِيّة، مثل: int و char، بل، وحتى المؤشرات pointers.

من أهم مزايا "سايثون": إمكانية استعمال مكتبات "سي" دون عمل وسيط لربطهما bindings، حيث إنك تستطيع استعمال header files بشكلٍ طَبْعِيّ؛ كما تفعل في لغة "سي". و"سايثون" هذه تطويرٌ ل pyrex.

١.مثالٌ ساذجٌ

سأقدم الآن مثالًا ساذجًا، وأقصد بذلك أنه يمكن عمله بالكامل في "بايثون"، دون الحاجة للغة "سي" أو "سايثون". لكن الفكرة هنا أنني سأقدم طريقةً للوصول لمكتبات "سي" من داخل "بايثون".

نريد عمل وحدةٍ للغة "بايثون"، تعمل على جمع ١٠٠٠ رقم عشوائيّ، بين ١ و ٦، وحساب وسطها الحسابي، مستخدمين الاستدعاء القياسي rand في لغة "سي"، من stdlib.h، ونريد استدعاء تلك الوحدة في لغة "بايثون".

إذا كتبنا man 3 rand؛ كي نقرأ كتيب rand، في لغة "سي"، كما يلي:

```
#include <stdlib.h>

int rand(void);
void srand(unsigned int seed);
```

التي يغلب استعمالها مع:

```
#include <time.h>

time_t time(time_t *t);
```

و time_t عبارة اسم آخر ل int.

والكود، في لغة "سي"، سيكون يشبه:

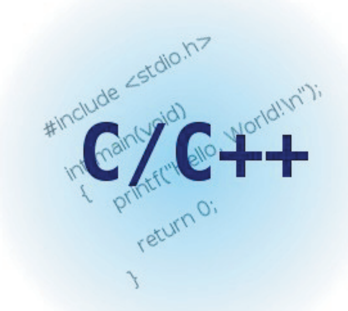
```
#include <stdlib.h>
#include <time.h>
double
rand_mean()
 {
  int sum=0,i;
  srand(time(0));
  for (i=0;i<1000;++i)
    {
      sum+=rand()%6+1
    }
  return sum/1000.0
 }
```

لكن، لتحويل هذه الوظيفة إلى وحدةٍ بلغة "بايثون"، يلزمنا الكثير من العمل. سندع "سايثون" تقوم به:

```
# myrand.pyx
cdef extern from "stdlib.h":
  int rand()
  void srand(unsigned int)

cdef extern from "time.h":
  int time(int *)

def rand_mean():
  """return the mean of 1000 numbers between 1-6"""
  sum=0
  srand(time(NULL));
  for i in range(1000): sum+=rand()%6+1
  return sum/1000.0
```



الآن، نكتب في سطر الأوامر cython myrand.pyx فنحصل على ملف myrand.c.

وأترك لكم مشاهدة ذلك الملف الناتج؛ لمعرفة ما كان يجب أن نكتبه؛ لنعمل تلك الاستدعاءات في لغة "سي" متوفرةً على شكل وحدةٍ للغة "بايثون".

الآن، نعمل ملف setup.py؛ كي يريحنا من عملية تصنيف ذلك البرنامج:

```
# setup.py
from distutils.core import setup
from distutils.extension import Extension
from Cython.Distutils import build_ext

ext = Extension(
    "myrand",                  # name of extension
    ["myrand.pyx"],            # filename of our Pyrex/Cython source
    language="c",              # this causes Pyrex/Cython to create C++ source
    cmdclass = {'build_ext': build_ext}
    )

setup(
  name = 'myrand',
  cmdclass = {'build_ext': build_ext},
  ext_modules = [ext],
)
```

الآن، نكتب في سطر الأوامر:

```
[alsadi@ojubabox cython]$ python setup.py build_ext --inplace 2>&1
```

كي تتم عملية تصنيف الوحدة التي صنعناها حيث سينتج ملف myrand.so.
الآن، شغل لغة "بايثون" في ذلك الدليل (المجلد):

```
[alsadi@ojubabox cython]$ python
Python 2.5.1 (r251:54863, Jun 15 2008, 18:24:51)
[GCC 4.3.0 20080428 (Red Hat 4.3.0-8)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import myrand
>>> help(myrand)
Help on module myrand:
NAME
    myrand

FILE
    /home/alsadi/cython/myrand.so

FUNCTIONS
    rand_mean(...)
        return the mean of 1000 numbers between 1-6
>>> myrand.rand_mean()
3.4159999999999999
```

بعض الشرح

الأمر cdef يكافئ def، التي تُستَخدم في الإعلان عن وظائف "بايثون"، لكنها هنا تستخدم في الإعلان عن نماذج prototypes لوظائف مكتبات "سي" الخارجية، التي نريد استيرادها. ونذكر في هذا الإعلان اسم ملف header file، الذي يحتوي على نماذج المكتبة، ولأن myrand لا تستقبل ولا تعيد بيانات غير بايثونية؛ لذا يمكن استدعاؤها من خارج "سايثون"، أما لو كانت تستقبل أو تعيد بيانات غير بايثونيةٍ؛ فإنها لا تكون مرئيةً خارجَ "سي" أو "سايثون".

أما ملف setup.py فهو ما يقابل ال Makefile في "سي". ولتضاف مكتبتنا التجريبية myrand.so بشكلٍ دائم إلى python، يجب نقلها إلى دليل مكتبات "بايثون"، والناتج من تنفيذ:

```
[alsadi@ojuba cython]$ python -c "from distutils.sysconfig import get_python_
lib; print get_python_lib(1)"
/usr/lib/python2.5/site-packages
```

مثالٌ حقيقيٌّ

كنت أريد عمل مشروع "بلال لأوقات الصلاة" على لغة "بايثون"، ولأن مكتبة ITL، من "عرب-آيز"، غير متوفرة لتلك اللغة إلا من خلال SWIG، والتي غالبًا لا تكون مثبتةً على أغلب التوزيعات. لذا، سنرى كيف سنعمل ربطًا لمكتبة ITL في "بايثون":

```
# iprayer.pyx
# -*- coding: utf-8 -*-
"""
Copyright (c) 2008 Muayyad Saleh Alsadi<alsadi@ojuba.org>
    Released under terms on Waqf Public License.
    This program is free software; you can redistribute it and/or modify
    it under the terms of the latest version Waqf Public License as
    published by Ojuba.org.

"""
cdef extern from "itl/prayer.h":
   # data types
   ctypedef struct Date:
      int day,month,year
   ctypedef struct Location:
      double degreeLong, degreeLat, gmtDiff,
      int dst,
      double seaLevel, pressure, temperature
   ctypedef struct Method:
      double fajrAng,ishaaAng,imsaakAng,
      int fajrInv,ishaaInv,imsaakInv,round,mathhab,
      double nearestLat,
      int extreme, offset
      double offList[6]
   ctypedef struct Prayer:
      int hour,minute,second,isExtreme
```

<<<<تابع بقية الكود البرمجي في الصفحة التالية >>>>

```
   # functions
   void getPrayerTimes (Location*, Method*, Date*, Prayer*)
   double getNorthQibla(Location*)
   void getMethod(int , Method*)

import time
import os
import os.path

def get_iprayers(date=None):
      """takes a date type (YYYY,MM,DD) wich defaults to the current day,
      and return a list of 6 tuples like (HH,MM,SS,isDST, isExtreme) for the
5 daily prayers and sunrise time"""
      cdef Location l
      cdef Date d
      cdef Method m
      cdef Prayer p[6]
      if date==None: date=time.localtime()[0:3];
      t=time.mktime(date+(12,0,0,0,0,0))
      d.year,d.month,d.day=date
      datestring="%04d%02d%02d" % date
      # Mekka settings
      l.degreeLong=39.82
      l.degreeLat=21.43
      l.seaLevel=298
      l.gmtDiff=3
      getMethod(6, &m)
      m.mathhab=1
      m.extreme=5
      l.dst=0
      getPrayerTimes (&l, &m, &d, p)
      r=[(p[i].hour,p[i].minute,p[i].second,self.__conf['DST'],p[i].isEx-
treme) for i in range(6)]
      return r
```

شرح ما سبق

بداية الكود السابق هو تعريفٌ للوظائف الموجودة في مكتبة ITL، وهي لاتحتلف عن محتويات ملف itl/prayer.h .

أما بقية الكود فهي وظيفةٌ بايثونيةٌ، تستقبل بياناتٍ بايثونيةٍ، وتعيد بياناتٍ بايثونيةٍ؛ لهذا يمكن استدعاؤها من خارج Cython.

الوظيفة get_iprayers، تأخذ التاريخ على شكل مرتب يحتوي السنة والشهر والتاريخ، ويجوز إهماله، وتعيد مرتبًا يحتوي ٦ أوقاتٍ في كل منها مرتبٌ يحتوي الساعة والدقيقة والثانية والتوقيت الصيفي وهل هناك تقدير أقرب البلدان ...إلخ.

لتقزيم الكود كي يناسب المقالة؛ تم تثبيت المدينة على مكة. لاحظ استخدام أساليب لغة "سي" في الحصول على point بواسطة عملية &.

الآن نحول كود "سايثون" إلى كود "سي" بواسطة الأمر cython iprayer.py.

ثم نعمل ملف setup.py ليعمل تصنيفًا له. ستلاحظ عدم نجاح ذلك بسبب linking errors.
إليك الملف الصحيح.

```
# setup.py
from distutils.core import setup
from distutils.extension import Extension
from Cython.Distutils import build_ext

ext = Extension(
    "iprayer",                      # name of extension
    ["iprayer.pyx"],                     # filename of our Pyrex/Cython source
    language="c",                   # this causes Pyrex/Cython to create C++ source
    #include_dirs=[...],            # usual stuff
    libraries=['itl'],                   # ditto
    #extra_link_args=[...],              # if needed
    cmdclass = {'build_ext': build_ext}
    )

setup(
  name = 'iprayer',
  cmdclass = {'build_ext': build_ext},
  ext_modules = [ext],
)
```

الاختلاف هنا هو استعمال libraries=['itl'] التي تكافئ -litl عند تصنيف كود ال"سي".



طريقة إنشاء فيديو رسوم متحركة باستخدام
FFmpeg و ImageMagick و Inkspace

للكاتبة: أنوار سيدام



هل تريد إنشاء رسوم متحركة، ثم تقوم بتحويلها إلى ملف فيديو بعد ذلك؟
هناك إحدى السبل التي تمكنك من القيام بذلك دون الحاجة إلى استخدام حزمة جِمْب للرسوم المتحركة GIMP Animation Package أو GAP اختصارًا.
بالرغم من أن استخدام حزمة جِمْب هذه جيدة، لكن من الممتع أن تتعرف على طرق أخرى بديلة؛ إن كنت ترغب في المزيد من المرونة والتحكم في المقاطع الخاصة بك.

إذًا هيا لنبدأ.

المتطلبات

سوف تحتاج إلى البرامج التالية:
Inkspace - لإنشاء المقاطع. تستطيع أيضًا استخدام جِمْب، لكن هنا سنستخدم Inkspace.
ImageMagick Suite - لتحريك الرسوم.
FFmpeg - للتحول إلى فيديو.

ومن البَدَهيّ أيضًا أنك ستحتاج إلى إحدة توزيعات جنّو/لينوكس أو إحدى أنظمة التشغيل الأخرى المبنية على يونِكس، مثل: BSD و Solaris؛ وذلك لأننا سنستخدم مجموعة من أسطر الأوامر في الطرفيّة.

إنشاء المقاطع

يتمثل جزءٌ كبير من المرونة في هذه الطريقة في السيطرة المطلقة على المقاطع الخاصة بك، وأوصي باستخدام Inkspace لهذه الخطوة؛ لأنها تتيح لك قدرًا كبيرًا من المرونة في الرسم.

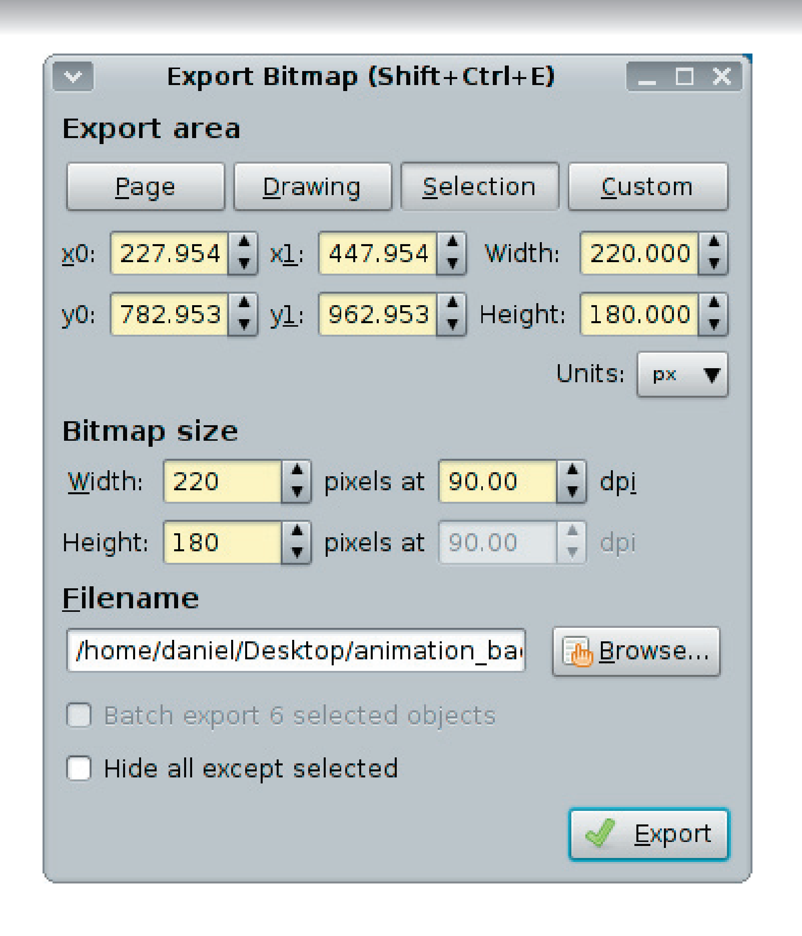
أولاً - نحتاج إلى تقدير حجم الخلفية التي نريدها للناتج النهائي للرسوم المتحركة.
فلنجعله هنا بمقاس ١٨٠×٢٢٠ بكسل.
ارسم مربعًا فقط، واختر قيم محوري العرض والارتفاع (X و Y على الترتيب). قم بإدخال ٢٢٠,٠٠٠ و ١٨٠,٠٠٠ على التوالي، وتأكد من إدخالهما بالشكل الصحيح.

يمكن جعل لون الخلفية بأي لون أو بأي درجة تريد. هذا أحد الأمثلة:



من الأفضل تصدير Export هذا المستطيل بوصفه أول شكل خاص بك للرسوم المتحركة، ومن الأفضل إنشاء مجلد خاص، وليكن "animation" أو أي اسم آخر حسب رغبتك؛ لتضع فيه كل ما تقوم بتصديره من صور. يمكنك أن تسمي هذه الصورة "٠" والأفضل أن تصدرها ”0.png“.

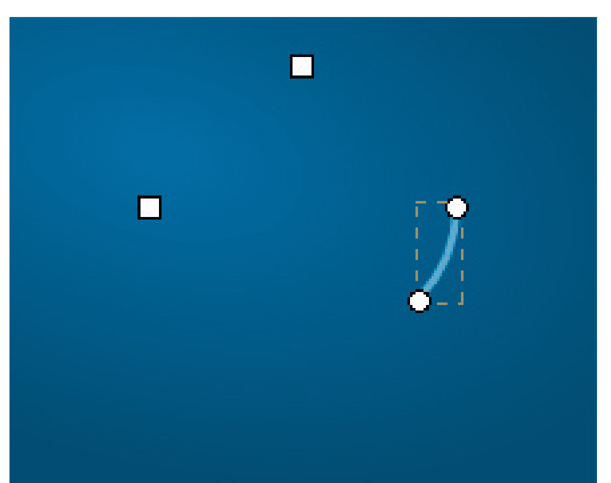
لتصدير صورة:
File > Export bitmap
ستظهر لك نافذةٌ حواريةٌ، اختر التبويب Selection، ثم انقر على Browse، الآن اختر المجلد الذي تود حفظ الصور فيه -من الممكن أن يكون المجلد "animation" المذكور سابقًا-، واحفظ الصورة باسم ”0.png“، وبمجرد الانتهاء انقر الزر Export.





إنشاء الخلفية المستطيلة كانت الخطوة الأولى. أي عمل تقوم به داخل هذا المستطيل سيظهر عند عرض الرسوم المتحركة. الآن علينا القيام بعمل الشكل الذي نريد أن يكون في بداية عرض الرسوم المتحركة، والذي يمكن أن يكون نصًّا أو رسمًا أو أيّ شيءٍ آخر. يمكنك إنشاء المقاطع بواسطة الرسم والكتابة بنفسك، أو يمكنك استيرادها Import من رسوم سابقة.

لعمل رسوم متحركة هنا فلنبدأ بدائرة زرقاء بدون ملئها بالألوان، مع رسم حدود سميكة إلى حدٍ ما، هكذا:



عظيم!. الآن ما أود فعله هو جعل الدائرة تُرسم بنفسها. يمكننا فعل ذلك عن طريق اختيار الدائرة والتبديل إلى الأداة Edit paths by node (وهو الزر الثاني في الأدوات باليسار، أو بواسطة الضغط على مفتاح F2).
هذا سيسمح لنا "بإزاحة" الدائرة -إن صحّ التعبير- وجعل الجزء المطلوب يظهر: انظر إلى الصورة لتُكوّن فكرةً عمّا أعنيه.

الجزئية الصغيرة، التي قمنا بها الآن، وُضعت كبداية عند رسم الدائرة. حيث إنّ كل الدائرة غير مرئية ما عدا هذه الجزئية المُوَضّحة في الصورة. سنقوم الآن بتصدير هذه الصورة كما فعلنا مع الخلفية المستطيلة.
حتّى هذه اللحظة، وحتّى مع المقاطع الأخرى أيضًا، نحتاج إلى هذه الخلفية أثناء تصدير الأشكال؛ فكل شيءٍ سيُصدّر كجزءٍ منها.

أعتقد الآن أنه قد تكوّنت لديكم فكرة عما ستفعلونه.

في المقطع التالي، ستقوم بإظهار جزئية أكبر بقليل من الدائرة، وستقوم بتصدير الصورة باسم "2.png" ومن ثمّ سترسم جزئية أكبر من الدائرة مرة أخرى، وتُصدّرها باسم "3.png" وهكذا... حتى تصبح لديك دائرة كاملة.
أظهر جزئيةً صغيرةً من الدائرة خطوةً بعد خطوة؛ لتجعل الدائرة ترسم بسلاسة أكثر، ومن ثمّ ستصبح رسمًا متحرّكًا.
وهذا هو أساس طريقة إنشاء المقاطع باستخدام برنامج Inkspace؛ مقطع بعد مقطع، وبهذا تتكون الرسوم المتحركة.

لتجعل الأمر أسهل عليك، يمكن ترك النافذة الحوارية Export bitmap مفتوحةً بينما تقوم بالتصدير. فقط عليك بتغيير الرقم في اسم الملف، والنقر على زر Export في كل مرة تقوم بالتعديل فيها على الدائرة.

يمكنك أيضًا زيادة بعض الإضافات على الشكل ليصبح أكثر جمالاً. مثلاً، نصنع شكلاً مُتدرجًا، وذلك بفضل خصائص Inkspace للتلاشي، كما في الشكل التالي:



لم نُضف هذه الخلفية إلى لجعل الشكل أفضل وأكثر جمالاً فقط. ما علينا فعله الآن هو وضعها فوق الدائرة الكاملة لنحصل على هذا الشكل:



قم بتحديدها، واذهب إلى:
Object > Fill and Stroke
ثم قم بتحريك شريط التمرير Opacity قرب الصفر، بعدها قم بتصدير الشكل -كما فعلنا سابقًا- مرةً أخرى، واستمر في ذلك حتى يصل شريط التمرير إلى ١٠٠٪.

يمكنك أيضًا كتابة أي شيء تريده على الدائرة: مثلاً، سوف أكتب "linuxac".
كما يمكنك تنفيذ نفس تأثير التلاشي السابق على النص، أو يمكنك عمل تأثير الكتابة على آلة طابعة، وذلك عن طريق كتابة أول حرف "l"، وتصديرها، ثم "li"، ثم "lin" ثم "linu"، وهكذا...

وهذا هو الشكل النهائي الناتج عن كل الصور المُصدّرة:



الخطوة التالية هي تجميع كل الصور مع بعضها البعض؛ حتى نحصل على رسومٍ متحركة. وهذا في الواقع هو الجزء الأيسر من العمل.

١.جمع الأشكال معًا لتُشكّل رسمًا متحرّكًا:

افتح الطرفيّة Terminal، واذهب إلى المجلد الذي وضعت فيه المقاطع الخاصة بك؛ فمثلاً، إن كانت موجودة في المجلد "animation" فاكتب الأمر التالي:

```
$ cd animation
```

يفترض أنك قمت بتثبيت حزمة ImageMagick. كل ما عليك فعله الآن هو كتابة الأمر التالي في الطرفيّة:

```
animate -coalesce -delay 9 0.png 1.png 2.png 3.png 4.png 5.png 6.png 7.png
8.png 9.png 10.png 11.png 12.png 13.png 14.png 15.png 16.png 16.png 16.png
16.png 16.png 17.png 18.png 19.png 20.png 21.png 22.png 23.png 24.png 25.png
26.png 27.png 28.png 29.png 30.png 31.png 32.png 32.png 32.png 32.png 32.png
32.png 32.png 32.png 32.png 32.png
```

coalesce- : هذا الخيار يقوم بتجميع المقاطع مع بعضها.

delay- : وهذا يحدد مدة التوقف بين المقاطع، وبذلك يمكننا تحديد السرعة الفعلية للرسوم المتحركة.
كما في الأمر السابق: حددنا القيمة "٩". يمكنك أيضًا تجربة القيم الخاصة بك.

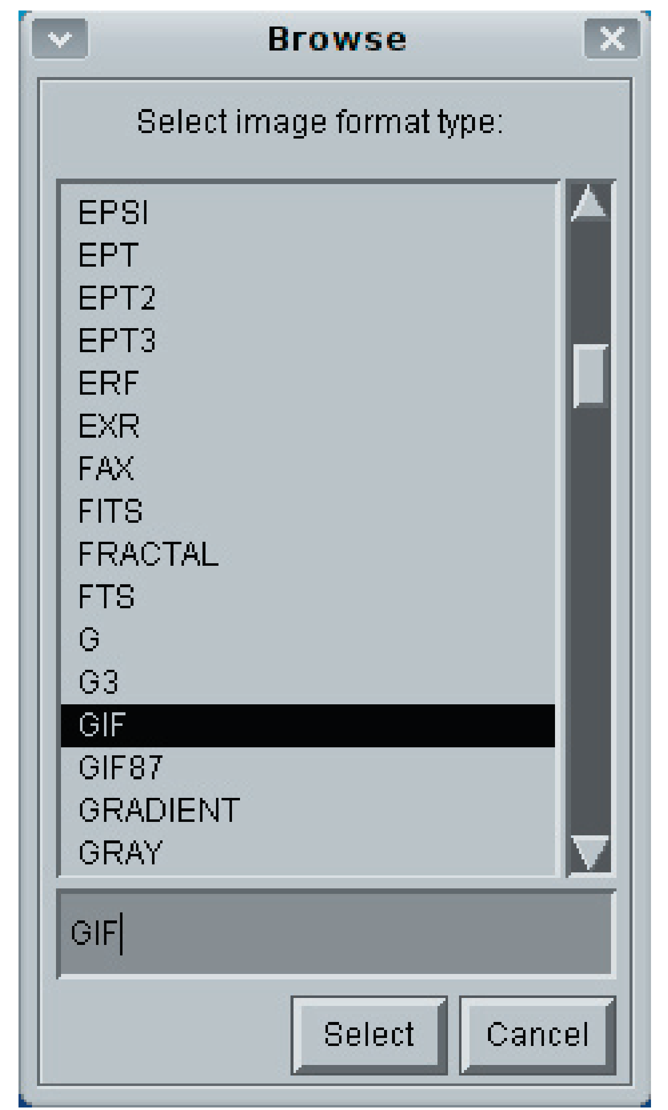
نكتب بعد ذلك أسماء ملفات الصور التي قمنا بإنشائها، ابتداءً من الملف "0.png"، حتى آخر صورة لديك "xx.png"، والتي كانت رقم "٣٢" في مثالنا هذا.

ملاحظة: ستجد أني كررت الصورة "png.١٦" خمس مرات، و "png.٣٢" عشر مرات.
تكرار الصورة أكثر من مرة يؤدي إلى طول مدة ظهورها، وهذا حسب عدد مرات تكرارها.

عندما يعمل معك سطر الأوامر بشكل جيد، انتظر قليلاً، ستظهر لك نافذةٌ بها الشكل النهائي للرسم المتحرك. لتحفظها بصيغة GIF: فقط انقر على النافذة وستظهر لك قائمة ImageMagick.



انقر على Animate، ثم اختر Save، ستظهر لك نافذةٌ أخرى لاستعراض المسار الذي تود حفظ الصورة فيه، ثم انقر على Format، ستظهر نافذة حوارية أخرى لاختيار الصيغة المناسبة.



اختر الصيغة GIF، ثم انقر على Select. تأكد من أن اسم الملف، والذي قمنا بتسميته "linuxac" ينتهي بـ ".gif"، ثم انقر على Save وستجد ملف الصورة المتحركة بصيغة GIF في المسار الذي قمت بتحديده.

التحويل إلى فيديو

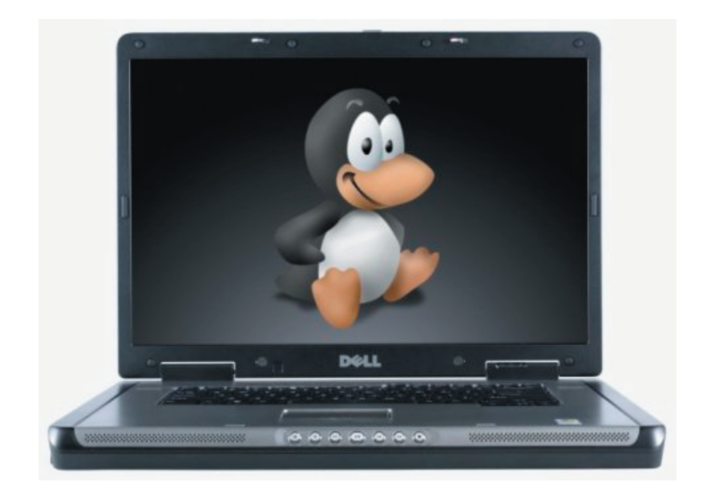
أخيرًا، وصلنا إلى نقطة التحويل إلى ملف فيديو. أولاً، علينا استخدام الأمر convert لتحويل GIF إلى JPEG، التي يستطيع FFmpeg تجميعها وتحويلها إلى ملف فيديو. من الأفضل نقل ملف الصورة GIF إلى مجلد آخر. قم بكتابة الأمر التالي في الطرفيّة:

```
$ convert linuxac.gif linuxac%05d.jpg
```

يمكننا الآن تشغيل برنامج FFmpeg لتجميع الصور وتحويلها إلى فيديو. مرةً أخرى، يمكنك استبدال "linuxac" بالاسم الخاص بك:

```
$ ffmpeg -r 12 -i linuxac%05d.jpg -y -an
linuxac.avi
```

الرقم ١٢ هذا يحدد معدل سرعة عرض الصور في الفيديو. يمكنك اختيار القيمة التي تريدها، لكني اخترت ١٢ لأنها تبدو لي الأنسب للفيديو الذي لدي.



الشبكات اللا سلكية وأساسيات حمايتها

للكاتب: علي الشمري



في السنوات الأخيرة، بدأت أسعار/تكلفة شراء معدات خاصة بالشبكات اللا سلكية تنخفض كثيرًا؛ بحيث أصبح اليوم بإمكانك عمل شبكةٍ لاسلكيةٍ متكاملة، سواءً في البيت أو العمل، بتكلفة بسيطة جدًا. كل ما ستحتاج إليهِ هو شراءُ Access Point أو AP، وفي الغالب هذا الـ AP يستعمل، في نفس الوقت، كـ Router. وأيضًا، سنحتاج إلى بطاقاتٍ خاصةٍ بالشبكات اللا سلكية تُرَكّب على الأجهزة التي نستعملها: سواءً كانت حواسيبَ مكتبيةً، أو حواسيب محمولة.

كل هذا جعل عملية تكوين شبكةٍ لا سلكيةٍ في بيوتنا وعملنا أسهل من السابق، وهذا ما حصل معي بالفعل؛ حيث قمت، منذ فترة قصيرة، بشراء جميع المعدات اللازمة لعمل شبكةٍ لا سلكيةٍ في المنزل؛ وذلك لكي أتخلص من كثرة الأسلاك الموجودة في المنزل، وأيضًا لغرض التعلم على الشبكات اللا سلكية، التي كما قلت، سابقًا، لم يكن بإمكاننا العمل عليها بسبب أسعارها المرتفعة، ولكن اليوم صارت بمتناول الأيدي، والحمد لله.

بعد شرائي لكل شيء، لم أقم بتشغيل الـ AP، وتشغيل الشبكة عندي في المنزل على هذه الشبكة، والسبب هو أنني خلال رحلتي الأخيرة إلى العراق؛ رأيت في الكثير من الأماكن شبكاتٍ لا سلكيةٍ غير محميةٍ، وبسهولةٍ تمكنت من اختراقها والاتصال بها واستعمال الـ BandWidth الخاص بها.

لهذا، قلت إنه يجب أن أتعلم كيف يمكنني حماية شبكتي قبل أن أقوم بتشغيل الشبكة وأجعلها مكشوفةً للعلن!.

بدأت بالبحث من خلال Google، صديق العائلة ، وبدأت بتعلم أساسيات الشبكات اللا سلكية، ومن ثم التعمق بها شيئًا فشيئًا. وبفضل الله بعد فترةٍ من الزمن تعرفت على الكثير من الأمور التي تمكنني، كبداية، من حماية شبكتي بأبسط صورةٍ ممكنةٍ، وبعد ذلك أبدأ في التطوير أكثر وأكثر.

أهم الأسباب التي جعلتني أتعلم حماية الشبكة اللا سلكية هي:

١.هذه الشبكة مكشوفةٌ للجميع، وباستطاعة أي شخصٍ، بأدوات معينة، وإمكانيات معينة، أن يتصل بها، ليس كما في الشبكات السلكية، التي يتصل بها، فقط، من لديه سلكٌ متصلٌ بها.

٢.شبكتي الداخلية هي ملك لي ولأهل بيتي، ولا أريد دخيلاً عليها.

٣.لا أريد أن يتم استغلال موارد شبكتي "للرايح والجاي".

٤.لا أريد أن يتم استهلاك الـ Bandwidth الشهري المسموح لي به من قبل شركة مزود خدمة الإنترنت من قبل شخصٍ من الخارج.

هذه هي أهم الأسباب، على الأقل، حسب وجهة نظري، لضرورة حماية شبكتي المنزلية، أو التي بالعمل، ولا أعتقد أن هذه الأسباب تهمني أنا فقط، بل هي مهمةٌ للجميع دون استثناء. هذا هو السبب الذي جعلني أكتب هذه المقالة البسيطة؛ التي سأحاول من خلالها توضيح أبرز الأمور التي ستحتاج إليها؛ للتخلص من الدخلاء على شبكتك اللا سكلية.

ربما يقول قائلٌ: شبكتي غير مهمة لكي يتم اختراقها من قبل أحد.

هذا الكلام غير سليم؛ وذلك لأن الشبكة من الممكن استغلالها لعدة أمور كالتي وضحتها سابقًا، لكن، بالإضافة إلى ذلك؛ فإنه يمكن استغلالها لشن هجماتٍ إلكترونيةٍ على شبكاتٍ أخرى، وبالتالي تصبح أنت المذنب والمسؤول عن مصدر هذه الهجمات، ولكن في الحقيقة لا دخل لك في أي شيء. أيضًا، عدم وجود معلوماتٍ مهمةٍ على شبكتك؛ فربما كان كل ما فيها مجرد ملفات عادية (صور، كتب، مرئيات، صوتيات، ...)، لا يعني هذا بأنها ليست محل اهتمام أحدٍ، وسأخبركم عن الذي حدث معي، منذ مدةٍ ليست بالبعيدة، في الشبكة اللا سلكية الموجودة عندي بالعمل.

كنت، كعادتي، أراقب الحركة Activity على الشبكة، سواءً السلكية أو اللا سلكية، وفجأةً وجدت حركةً غير عادية صادرة من جهاز أحد الموظفين وهو يقوم بتصفح مواقع لا أخلاقية، وليس هذا فحسب، وإنما المواقع التي يقوم بتصفحها عبارةٌ عن مرئيات، أي Streaming، كالتي في YouTube!. بعد الفحص والتأكد من الموظف؛ تبين أن التشفير المستعمل في الشبكة اللا سلكية من نوع WEP، وهذا التشفير بطبيعة الحال من السهل جدًا اختراقه، ولن تحتاج إلا لبضع دقائق للقيام بذلك، والكلمة السرية المستعملة، وقتها، كانت من ٥ خانات: حروف/رموز/أرقام فقط. هذه الشبكة تم تركيبها من قبل شركة مختصة بأمور الشبكات اللا سلكية، وفي وقتها لم تكن لدي لا خبرةً بالشبكات اللا سلكية ولا وقت فراغ كافٍ لتعلم أهم أمور الشبكات اللا سلكيةً. لكن كما قلت؛ فإني قد بدأت في القراءة والمتابعة، وتعلمت الكثير من الأمور التي تخص هذا النوع من الشبكات؛ ولهذا استطعت أن أعرف أين نقاط الخلل الموجودة في شبكتنا، وبدأت في إصلاحها، بحمد الله وفضله. لم يكن المخترق قد عرف اسم الـ Service Set ID، أو ما يسمى SSID وحسب!، أو قام بفك تشفير الكلمة السرية المستعملة فحسب، إنما قام بعمل Spoof على MAC Address لأحد أجهزة الموظفين واستعماله في الاتصال بالإنترنت مستغلاً شبكتنا المحلية، وبالتالي كانت كل الأدلة تشير إلى هذا الموظف المسكين عندنا في الشركة .



على ضوء ما ذكرته، وما حصل معي، سأخبركم بطريقةٍ بسيطة لحماية شبكاتكم المحلية وتقليل فرص إمكانية اختراقها؛ لأنه، وكما يعلمُ الجميعُ، لا توجد حماية ١٠٠٪ بتاتًا، والكمال غير موجودٍ في عالم البشر، إنما، فقط، لخالق البشر وحده سبحانه وتعالى.

بعد هذه المقدمة الطويلة، وربما المملة، أبدأ بأهم الخطوات التي أتمنى أن تقوموا بتطبيقها على الـ AP الموجود لديكم، وعلى أجهزتكم، لأن بعض الخطوات يخص الـ AP، وبعضها يخص أجهزتكم الشخصية نفسها. أيضًا، أود أن أخبركم بأن هذه الخطوات، جميعها مع بعضها البعض، ستشكل عوائق للمخربين وزيادة في الحماية، لكن ليس بالضرورة أن تقوم إحدى النقاط بحمايةٍ كاملةٍ لمسألةٍ معينةٍ؛ ولهذا من الضروري تطبيقهم جميعًا:

١.قم بتغيير اسم الـ SSID الأساسي؛ لأنه يمكن معرفة أسماء الـ SSID لأي نوع AP من خلال الإنترنت. أيضًا، نصيحةً، لا تقم باستعمال اسم شركتك أو اسمك؛ لكي لا تكشف هذه الشبكة لكل من هبّ ودبّ، ويبقى الاسم نوعًا من أنواع الـ Privacy لك.

٢.قم بإيقاف خاصية الـ **SSID Broadcast** على الـ AP؛ وذلك لكي لا يظهر اسم شبكتك في حال وجود أي جهاز قريبًا من مدار شبكتك، ويملك أجهزة اتصال لا سلكية؛ أي إن الهدف هو جعل الـ SSID مخفيًا. لكن، لأكون صادقًا معكم؛ فإن أي شخصٍ، وباستخدام بعض الأدوات، يستطيع أن يكتشف الـ SSID الخاص بأية شبكةٍ مخفيةٍ، ومثالٌ على هذه الأدوات: kismet.

٣.قم بتغيير اسم المستخدم وكلمة السر الخاصة بالوصول إلى الـ Access Point. اللذان يطلبهما منك عند محاولة الدخول إلى صفحة إعدادات الـ AP من خلال المتصفح. قم بتغيير هذا الاسم، وقم بوضع كلمةٍ سريةٍ قويةٍ له.

٤.قم باستعمال التشفير Encryption في الشبكة، ولا أنصح بتاتًا باستعمال البروتوكول WEP؛ حيث إنه بروتوكول قديم، ويسهل كسره بسهولة، والإنترنت تعج بطرق كسره واختراق الشبكات المحمية به. قم باستعمال WPA-PSK، واستعمل التشفير من نوع TKIP. باستعمالك لتشفير TKIP سيتيح لك استعمال Passphrase أي عبارة للمرور طولها من ٠ إلى ٦٣ رقم/رمز/حرف، وكلما كانت طويلةً ومعقدةً (خلط بين الأنواع الثلاثة) ستكون صعبةً ومعقدةً على المخترق من أجل كسرها. اختراق هذا النوع من البروتوكولات ليس بنفس سهولة اختراق WEP، وأيضًا، هذا النوع يعتمد في اختراقه على كلمات الـ passphrase الضعيفة؛ وذلك لأنه يتم اختراقه من خلال طرق الـ Brute Force فقط. هناك أدوات تستعمل لاختراقه بطريقة Brute Force، مثل: coWPAtty، و aircrack.



٥.قم بتشغيل خدمة الـ MAC Address Filtering على الـ AP، وقم بإضافة الـ MAC Address الخاص بالأجهزة التي تريد السماح لها باستعمال شبكتك اللا سلكية. بالطبع، هذه الطريقة يمكن تجاوزها من خلال معرفة MAC Address لجهازٍ متصل على هذه الشبكة، وبالتالي تقوم بتغيير الـ MAC Address الخاص بجهاز المخترق إلىً ذلكً الـ MAC Address، وهكذا، أصبحت ذا صلاحيات على هذه الشبكة. لكن بدون معرفة كلمة المرور؛ هذا الـ Spoof لن يصبح ذا قيمة مهمة. علىً كل حالٍ، الأجهزة التي بمتناول أيدينا، التي ذكرت أنها اليوم بأسعار رخيصة، لا تملك ميزاتٍ لاكتشافٍ مثل هذه الأنواع من الاختراقات، وأقصد MAC Address Spoofing،

لكن الأجهزة المتطورة تملك أنظمة Intrusion Detection System مبنية عليها (AP) إحدى وظائفها هي اكتشاف الـ MAC Address Spoofing.

٦.قم باستعمال Static IP على الأجهزة التي تريدها أن تتصل بالشبكة اللا سلكية، وقم بربط هذه العناوين مع الـ MAC Address لكل جهاز على الـ AP؛ وذلك لكي تزيد من تعقيد الاتصال بشبكتك أكثر وأكثر. هذه نقطةٌ ضروريةٌ فلا تنساها.

٧.بعض الناس يعتقد أن المكان الذي قمت بوضع الـ AP فيه غير مهم، وهذا كلام غير سليم؛ فإن من أهم النقاط هو مكان وجود الـ AP، ليس فقط لكي يوصل الإشارة إلى جميع من في المنزل/العمل، وًإنما لكي تحد من مشاكل اتصال الآخرين بالشبكة. قم باختيار أفضل مكان ممكن للـ AP بدون أن يؤثر على قوة الإشارة الموصلة لكل أرجاء المنزل/العمل.

٨.قم بتركيب جدارٍ ناريّ على كل جهازٍ مرتبطٍ بالشبكة اللا سلكية، وبطبيعة الحال حتى السلكية. هذه نقطةٌ مهمةٌ جدًا؛ لزيادة الحماية أكثر وأكثر، وتعقيد المُخترق، وجعلهُ يمل منَ المحاولة؛ فلا يملك إلا أن يفر من عندك بعد كل هذه التعقيدات.

٩.قم بإغلاق الشبكة اللا سلكية في الأوقات التي لا تُستعمل فيها. فلماذا تُبقي الشبكة تعمل إذا لم يكن يوجد من يستخدمها؟!. الأفضل أن تقوم بإغلاقها. بالطبع، هذه النقطة ربما غير ممكنة في الأماكن التي بها ضغط عملٍ، كما في الشركات، ولكن، ربما في المنزل هي مفيدةٌ، خاصةً أنك لن تكون مراقبًا لشبكتك في تلكً الأوقات.

١٠.أحببت أن أؤكد لكم أن هذه النقاط جميعها مع بعضها البعض تزيد من الحماية الموجودة لديك على الشبكة، ولكنها لا تحميك بنسبة ١٠٠٪؛ فالكمال غير موجود، والحماية لن تصل يومًا إلى النسبة الكاملة.

في الختام أتمنى أن تكونوا قد استفدتم من هذه المقالة البسيطة، بإذن الله.



مقدمة إلى الـ Rootkit

للكاتبة: روضة الصوابني



إن حماية نظام التشغيل من محاولات الاختراق أصبح حاجةً أساسيةً لا غنى عنها، ومن هنا تظهر الحاجة إلى ضرورة إلمام المستخدم بالمخاطر التي يمكن التعرض إليها.
من هذا المنطلق؛ سنحاول دراسة كيفية اختراق نظام التشغيل باستعمال برامج الـ Rootkit، وطرق الحماية.

ماهو الـ Rootkit؟

الـ Rootkit هو عبارة عن مجموعة أدوات يستعملها المخترق الذي تمكن من اجتياح نظام التشغيل بنجاح؛ لجعل مدير النظام يجهل وجوده، وللحصول على صلاحياتٍ تُخَوِّل له تنفيذ برامج وسكريبتات على جهاز العميل.Client

لمحةٌ تاريخيةٌ عن الـ Rootkit

سُمي الـ Rootkit بهذه التسمية؛ لأنه يُتيح للدخيل أن يصبح المستخدم الجذر (مدير النظام) بالنسبة لنظام التشغيل "لينوكس"، ومنذ ذلك الحين تم تطوير برمجيات مماثلة لأنظمة التشغيل الأخرى. تم توسيع مصطلح الـ Rootkit ليشمل أي برنامجٍ قادرٍ على أن يغير، خلسةً، نظام التشغيل بحيث يمكّن للمستخدم الغير مأذونٍ له أن يدير رقابةً تعسفيةً للنظام.

كُتب أول Rootkit معروف حوالي سنة ١٩٩٠، بواسطة لين ديفيس و رايلي ديك لـ Sun OS 4.1.1. في ذلك الوقت، انتبه مستخدمو نظام التشغيل للسلوك الغريب للخادم: كفقدان مساحة القرص، والاختفاء الغريب لدورات وحدة المعالجة المركزية، واتصالات الشبكة عند تنفيذ الأمر netstat.

أصبح الـ Rootkit معروفًا على نحو أفضل بكثير في عام ٢٠٠٥ عندما تسبب Sony BMG في خطأ فادح تمثل في إدراج برنامج Rootkit على أقراص مدمجةٍ للموسيقى. قامَ هذا الأخير بالعبث في نظام التشغيل "ويندوز"؛ لإتاحة وصولٍ أيٍ شخصٍ على بينة من تركيب الـ Rootkit للنظام.

ما هي أنواع الـ Rootkit؟

١. Binary Rootkits (الثنائيات):
أول Binary Rootkit أُستخدم ليحل محل ثنائيات النظام الحرجة والهامة مثل: /bin/login/ و network daemons.
استخدم المهاجمون هذا النوع من الـ Rootkit؛ لتحقيق عدة أهداف، منها النفاذ عن بعد، النفاذ المحلي وإخفاء الأدلة.
أول Rootkit كان عبارة عن إنشاء محفوظات (tar Archives) للعديد من ثنائيات النظام المتداولة؛ التي يتم تشغيلها من طرف المستخدم الجذر للتحقق من سلامة النظام.

٢. Kernel Rootkits:
هو أكثر الأنواع انتشارًا اليوم، والموجود على مستوى النواة.
أول Kernel Rootkit كان مُبرمَجًا لنظام "لينوكس" (منتصف١٩٩٧)، لكن سرعان ما تحولت هذه التقنيات إلى أنظمة التشغيل الأخرى "سولاريس" (١٩٩٩)، وأنظمة "بي آس دي" BSD الحديثة (١٩٩٩).
بالنسبة لهذا النوع: تنتقل الشفرة الخبيثة إلى النواة مثلما تنتقل وحدات النواة القابلة للتحميل باستخدام الطرق المنصوص عليها من قبل نظام التشغيل؛ لتحميل السائقين Drivers وقت التشغيل.

٣.Library Kits (مجموعات المكتبة):

مجموعات مكتبات حصان طروادة Library Trojan Kits. واحدٌ من أشهر ممثليها هو T0rn8 .

تستخدم أساليب مختلفة، يُستعصى الكشف عنها؛ فعلى سبيل المثال: تستخدم مجموعة T0rn8 مكتبة نظام خاصة تدعى libproc.a تحل هذه المكتبة محل المكتبة القياسية للنظام، والمستخدمة لعملية نقل المعلومات من مجال النواة Kernel Space إلى حيز مرافق Utilities المستخدم مثل /bin/ps/.
عندما تستقر المكتبة الطروادية بالنظام؛ لا يُسمح لأحدٍ بتعديل الثنائيات، حيث إنها سوف تستخدم البيانات من قبل libproc.a. على سبيل المثال: تستطيع هذه المكتبة إخفاء بعض أسماء الإجراءات، لاكتشاف ذلك يمكن النظر إلى /proc/.

٤.Firmware Rootkits:

يستخدم هذا النوع الجهاز Device أو المنصة Platform؛ لخلق صورة دائمة للبرامج الضارة، ويمكن لهذا النوع من الاختباء بنجاحٍ في الـ Firmware، وذلك لأن هذا الأخير لا يخضع إلى معطيات تكامل الشّفرة Code Integrity .



٥.Virtualized Rootkits:

تعمل هذه النوعية عن طريق تعديل سلسلة الإقلاع للجهاز، وذلك لتحميل نفسها بدلاً عن نظام التشغيل الأصلي. حينما يصبح الـ rootkit مُحمّلاً بالذاكرة؛ يقوم بتحميل نظام التشغيل الأصلي كجهازٍ ظاهريّ؛ مما يمكنه من الاعتراض لجميع نداءات العتاد لنظام التشغيل الضيف.

كيف تكتشف ما إذا تم تركيب Rootkit على جهازك؟

سنقوم في هذه المرحلة بشرح اثنين من البرامج التي تساعد على اكتشاف ما إذا كان تم تركيب Rootkit على جهازك، وهي: Root Kit Hunter و chkRootkit.

١.برنامج Root Kit Hunter:

يقوم هذا البرنامج (Script) بالتحقق من والكشف عن قرابة ٥٨ من الـ Rootkits، ويقوم بالتأكد من الخدمات والمنافذ المفتوحة بالخادم Server، ويقوم بإعطائك تحذيرات أمنية، لأي شيء قد يشكل خطورة على الخادم، والكثير من الميزات الأخرى.
أثناء الفحص يقوم البرنامج بتشغيل سلسلةٍ من الاختبارات التي تحقق في الملفات الافتراضية التي يستخدمها Rootkits، والأخطاء على مستوى تصريحات بعض الملفات وفحص وحدات النواة وما إلى ذلك.
يمكن تحميل البرنامج من موقع Sourceforge.net، أو عن طريق مدير حزم Synaptic بالبحث عن حزمة rkhunter بالنسبة للإصدارات التي تدعم Synaptic.
وهذه طريقةٌ أسهل لتركيب Root Kit Hunter على جهازك، وهي منقولة من مجتمع "لينوكس" العربي:
قم بفتح الطرفيّة Terminal، والدخول بحساب المُستخدم الجذر root، ثم اكتب الأمر التالي، لتنزيل Donwload ملف البرنامج المضغوط:

```
wget http://downloads.Rootkit.nl/rkhunter-1.2.7.tar.gz
```

ثم الأمر التالي لفك الضغط عن الملف:

```
tar -zxvf rkhunter-1.2.7.tar.gz
```

للدخول إلى المجلد الذي تم فك ضغط الملفات فيه؛ نطبق الأمر التالي:

```
cd rkhunter-1.2.7
```

وبعدها مباشرة الأمر التالي؛ لتنصيب البرنامج:

```
./install.sh
```

لبدء الفحص Scan؛ اكتب الأمر التالي:

```
rkhunter -c
```

ولتحديث البرنامج:

```
rkhunter –update
```



٢.برنامج ChkRootkit:

هذا برنامجٌ آخر لفحص حاسوبك.
وهو، مثل برنامج Root Kit Hunter، يُمكن إضافته عن طريق مدير الحزمSynaptic، أو يكفي تحميل الملف المضغوط، وفتحه، ثم تنفيذ الأمر chkRootkit.

البرنامج يقوم بإجراء سلسلة من الاختبارات على عدد من الملفات الثنائية، ومثل البرنامج السابق، يقوم chkRootkit بفحص تصاريح الملفاتً والعديد من الاختبارات الأخرى.

لمعرفة خصائص البرنامج؛ نكتب الأمر:

```
chkRootkit -h
```

لسرد قائمة الاختبارات التي ستجري على نظامك؛ اكتب الأمر:

```
chkRootkit -l
```

للقيام بفحص جهازك؛ اكتب الأمر التالي:

```
chkRootkit
```

وإذا كانت لديك خبرة متقدمة في نظام التشغيل "لينوكس"؛ يمكنك إضافة الخاصية X:

```
chkRootkit -x
```

Root Kit Hunter و ChkRootkit يشكلان معًا أداةً جيدةً لكشف الـ Rootkits على نظام "لينوكس".

كيف تزيل الـ Rootkit من جهازك؟

تتمحور عملية إزالة الـ Rootkit حول نقطتين، هما:

- إزالة الـ Rootkit نفسه: تسبب هذه العملية بعض المشاكل؛ وذلك نظرًا للتغيرات التي يحدثها الـ Rootkit بنظام التشغيل. يمكن أن تخلف عملية إزالة الـ Rootkit مشاكل لنظام التشغيل، كأن يصبح هذا الأخير غير مستقر، أو غير وظيفي.

- إزالة البرامج الضارة الخفية: تتعرض هذه العملية للمشاكل العادية التي تواجه عملية إزالة أي برنامج ضار، لكن، لن نتمكن من القيام بذلك حتى تتم إزالة الـ Rootkit، وعند هذه النقطة، كما أسلفنا الذكر، يمكن للنظام برمته أن يصبحَ غيرَ مستقرٍ إلى حدٍ أنه يصعب إزالة البرامج الضارة إزالةً تامة.

إذا كان المستخدم واثقًا من وجود صورةٍ سليمة لمحرك الأقراص -أنشأت قبل الإصابة بالـ Rootkit-؛ فيمكنه استعادة محرك الأقراص، وهكذا يمكن لبرنامج التصوير استعادة قطّاع الإقّلاع للقرص(١).

كيفية الوقاية من الـ Rootkits؟

لا تختلف الوقاية من الـ Rootkits عن الوقاية من أي نوعٍ آخر من الهجمات، فكلها تبدأ بتوفير البنود الأساسية لأمن نظام التشغيل.

تتكون المنظومة الأمنية لنظام التشغيل أساسًا من البنود التالية:

- جدار ناري(٢):
إن استخدام جدارٍ ناريٍّ، دائمًا، يُعد ممارسة جيدة؛ للتأكد من أن جميع الشبكات محمية من الإنترنت.

- التعرف بالضبط على كل ما هو بصدد العمل على نظام تشغيلك:
بعد تركيب النظام، ضع قوائم جردٍ لما هو قيد التشغيل، وإيقاف تشغيل الخدمات التي لا حاجة لها، وأداء مراجعاتٍ دورية لنظام العمليات؛ لضمان عدم وجود تطبيقاتٍ أخرى، غير المأذون لها، قيد التشغيل.

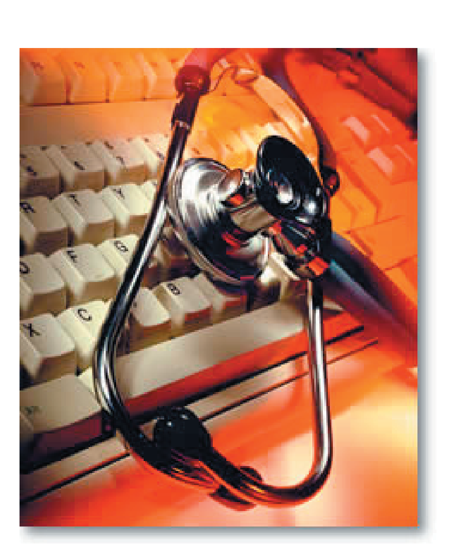
- تحديد صلاحيات كل مستخدم بدقة:
انتبه من أن تعطي المستخدم صلاحيات للوصول إلى خدمات الشبكة أكثر من احتياجاته. فقط الصلاحيات التي تخول له أداء وظيفته.

- استعمال اتصالات آمنة مثل VPNS و Secure Shell:
من خلال تنفيذ VPNS؛ يمكن التأكد من تشفير البيانات المرسلة عبر الشبكة، وبذلك تضمن سلامتها. استعمال ssh كبديلٍ لبروتوكول Telnet؛ فهو يقوم بتشفير البيانات المرسلة عبر الشبكة بما فيها أسماء المستخدمين وكلمات المرور.

- القيام بآخر التحديثات:
تجربة و تطبيق كل التحديثات الخاصة بأمن النظام.
استعمال تحديثات موثوق بها كالتي توفرها المواقع الرسمية للتطبيقات أو أي موقع موثوق به، وهنا ننبه خاصة إلى مقرات الحزم التي يتم إضافتها دون التأكد من مدى سلامتها.

- رصد جميع ملفات الدخول:
رصد ملفات الدخول يعطي صاحب النظام اليد العليا لمراقبة ماذا جرى في النظام؛ حيث إن أغلب الأنشطة يتم تسجيلها. إنها فكرةٌ جيدةٌ أن تجعل ذلك آليًا، باستخدام برامج فحص الدخول مثل logwatch، أو logsentry. هذه البرامج لن تحميك، حقيقةً، من الـ Rootkit، لكنها ستنبهك للأنشطة الغير عادية: كمحاولةٍ فاشلة لتسجيل الدخول.
تنفيذ الاحتياطات الأمنية الأساسية هي الخطوة الأولى لحفظ النظام من المُتسللين، ومنع أي نوعٍ من أنواع الهجمات. يمكن أيضًا منع الهجمات التي يشنها Rootkit بضمان أمن الملفات.

- سمة الملف الأمني:
التأكد من أنه لا يمكن تغيير الملفات المشتركة، ويمكن تحقيق ذلك عن طريق تحديد علم Flag ثابت على الملفات المهمة.



- استعمال العلم "غير قابل للاستخدام":
باستعمال العلم "غير قابل للاستخدام" على ملف ما؛ تضمن عدم تعرض هذا الأخير إلى التغيير أو الحذف أو إعادة التسمية أو حتى الوصلٍ إليه.

لتعيين علم "غير قابل للاستخدام" على ملفٍ؛ استخدم الأمر "chattr" الموجود في معظم توزيعات "لينوكس".

لتعيين العلم "غير قابل للاستخدام"؛ اكتب الأمر:

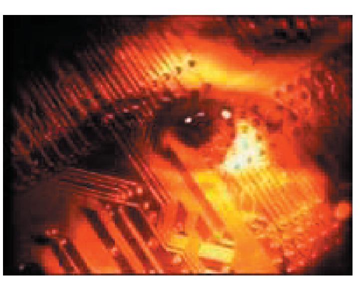
```
chattr +i <file >
```

لحذف علم "غير قابل للاستخدام"؛ اكتب الأمر:

```
chattr -i <file>
```

لعرض جميع صفات الملفات؛ اكتب الأمر التالي:

```
lsattr <file>
```



استخدام العلم "غير قابل للاستخدام" يعين على إفشال عمل بعض الـ Rootkits بما أن الملف المستهدف غير قابل للتغيير. لكن هذه الطريقة لا تعمل مع الـ Rootkits التي تعمل على مستوى النواة؛ فإنه لا يمكن تحديد صلاحيات عدم التغيير في هذا المستوى.

لا يمكن تعيين علم "غير قابل للاستخدام" إلا باستعمال حساب المستخدم الجذر. تذكر أيضًا أن العلم يمكن إزالته بسهولة؛ فمعنى ذلك أن المهاجم إذا استطاع الوصول إلى الآلة؛ فإنه سيكون قادرًا على إيجاد العلم و إزالته. لا تعتمد على علم "غير قابل للاستخدام" وحده كوسيلةٍ للدفاع ضد الـ Rootkits.

كيف يعمل الـ SucKit Rootkit؟

تم نشر وشرح كيفية عمل الـ SucKit Rootkit سنة٢٠٠١، ولقد قمنا باختياره في هذه الفقرة نظرًا لسهولة استعماله، ولأنه ينتمي إلى مجموعة الـ Rootkit التي تعمل على مستوى نواة "لينوكس"، والتي تعتبر الأكثر انتشارًا.

من مميزات هذا الـ Rootkit أنه يحتوي على آليةٍ لإعادة تحميل نظام التشغيل ومستتر (٣) Backdoor ينشط بعدما يتم إرسال بعض الرزم Packets إلى النظام عبر الشبكة.

يتم تفعيل الـ SucKit Rootkit كالآتي:

يقوم نظام التشغيل "لينوكس" بتنفيذ الأمر sbin/init/ عند بداية تشغيله.
SucKit يستبدل هذا الملف بمحمّله الخاص.
المحمّل يحقن الـ Rootkit في النواة، وينفذ أمر init الأصلي.



وحتى يضمن عدم اكتشافه؛ يقوم SucKit بإخفاء ملف init الأصلي.
بالطبع أية محاولة وصول إلى sbin/init/ يعاد توجيهها إلى الملف الأصلي، والذي تم إخفاءه.

كيفية حقن الـ Rootkit في النواة:

يقوم الـ Rootkit بنقل نفسه إلى النواة باستعمال dev/kmem/.
و هذه الطريقة أكثر تعقيدًا من استعمال وحدات النواة Kernel Modules، لكنها تتميز بصعوبة تجميدها والتصدي إليها.

يتم حقن شفرة الـ Rootkit في النواة عبر عدة مراحل:
يتم البحث في ذاكرة النواة على عنوان جدول نداءات النظام syscall وعنوان الوظيفةkamalloc.

تعتبر kamalloc وظيفةً داخليةً للنواة، وتحتاج لحجز مجالٍ في ذاكرة النواة.
يوضع عنوان kamalloc في مدخلة غير مستعملة من جدولٍ نداءات النظام.
يتم تنفيذ kamalloc كنداء نظاميّ، ويحجز تباعًا مجالاً من ذاكرة النواة.
يتم كتابة الـ Rootkit في المجال الذي تم حجزه من قبلkamalloc في النواة.
يوضع عنوان الـ Rootkit في مدخلة غير مستعملة من جدول نداءات النظام وبالتحديد تتم إعادة الكتابة على عنوانkamalloc.
يتم نداء الـ Rootkit كنداءٍ نظاميّ، ويكون في وضع تشغيل على مستوى النواة.

مخطط تفعيل الـ SucKit Rootkit:

حيث إن الـ SucKit Rootkit تتفاعل مباشرةً مع جدول نداءات النظام syscall؛ فإنها تقوم أولاً بنسخ نسخةٍ من هذا الجدول، وبهذه الطريقة تضمن البقاء على الجدول الأصلي بمنأى عن أي تغيير؛ مما يُضلل الأدوات التي تهدف إلى الكشف عن تناسق موارد النواة.

يتم بعد ذلك التلاعب بمعالج الانقطاع (٤) الخاص بنداءات النظام لاستعمال النسخة الخبيثة من جدول النداءات.

هوامش:

(١) قطاع الإقلاع: Boot Sector أو "قطاع البدء" هو قطاعٌ على القرص الصلب أو القرص المرن أو أية وسيلة تخزين بيانات مشابهة. يحتوي هذا القطاع على شفرةٍ تقوم بتنفيذ نظام التشغيل المخزن على أجزاءٍ أخرى من تلك الأقراص.
http://ar.wikipedia.org/wiki/قطاع_الإقلاع

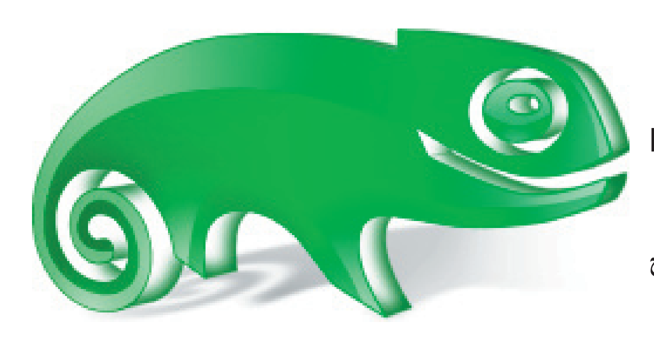
(٢) الجدار الناري: يسمى أيضًا بجدار اللهب Firewall. هو جهازٌ و/أو برنامجٌ يفصل بين المناطق الموثوق بها في شبكات الحاسوب، ويكون أداةً مخصصةً أو برنامجًا على جهاز حاسوبٍ آخر؛ الذي بدوره يقوم بمراقبة العمليات التي تمر بالشبكة، ويرفض أو يقرر أحقية المرور ضمنًا لقواعد معينة.
وظيفة الجدار الناري من داخل الشبكة مشابة لأبواب الحريق في تركيب المباني؛ ففي الحالة الأولى يستعمل في منع اختراق الشبكات الخاصة، و في الحالة الثانية يفترض أنه يحتوي ويؤخر الحريق الموجود في بناءٍ معينٍ من الانتقال إلى بناء آخر
http://ar.wikipedia.org/wiki/جدار_ناري

(٣) المستتر: Backdoor، هو بديلٌ للدخول إلى جهاز الحاسوب، والسيطرة عليه عن بعد.

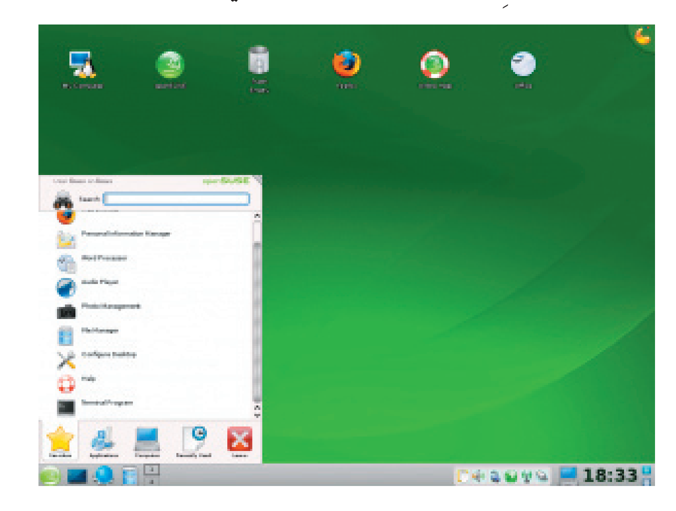
(٤) معالج الانقطاع: Interrupt Handler. يعالج الانقطاعات المؤقتة لعمل برنامجٍ ما على المعالج Processor بسبب استخدام هذا الأخير من قبل برنامجٍ آخر.

تثبيت أوبن سوزه ١١ على ذاكرة حية خارجية Live USB

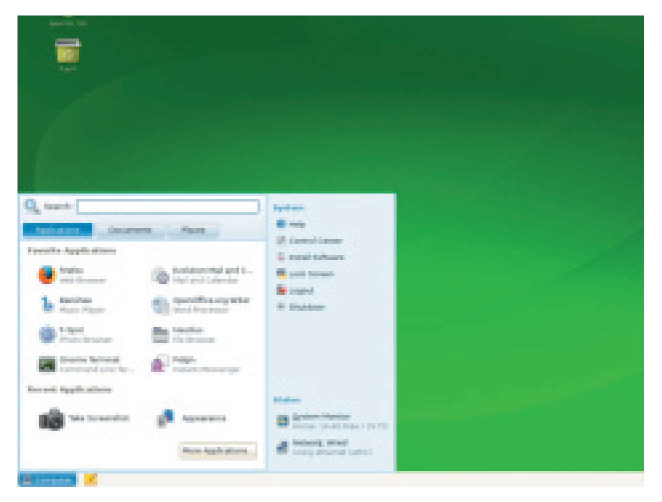
للكاتب: أبو بكر الطليبي



من منّا لم يسمع بنظام الحرباء أو "أوبن سوزه"؟
بالتأكيد كلنا يعرفها، وهي تأتي على شكلين: إما توزيعة تنصيبية Installable أو توزيعة حيّة Live CD.
لكن اليوم، سنجعلها على شكلٍ ثالث، الذي هو Live USB.
بعد هذا الموضوع سيصبح لدينا نظام "أوبِن سوزه" قادرٌ على الإقلاع من مفتاح USB.



أوبِن سوزه بالواجهة كيدي



أوبِن سوزه بالواجهة جنوم

نقوم بتحميل إحدى هاتين النسختين، ولا مشكلة بين نسخة الواجهة كيدي ونسخة الواجهة جنُوم.

يمكننا عمل الخطوات التالية باستخدام أي من نظامي التشغيل: "ويندوز"، أو "لينوكس"... لنا الخيار.

باستخدام نظام التشغيل "ويندوز"

سنستخدم البرنامجين: UltraISO و syslinux.
نجهز مفتاح USB بسعة تخزينية لا تقل عن ١ جيجا بايت.

- ننسخ محتوى التوزيعة باستخدام برنامج UltraISO، ونلصقه في مسار مفتح USB.
- نفك الضغط عن برنامج syslinux في القسم C، ثم نفتح موجه الأوامر Command Prompt.
- نطبق الأمر التالي:

```
cd c:\syslinux-3.71\win32
```

- ثم الأمر التالي:

```
syslinux -ma E
```

مع ملاحظة أن الحرف E يقصد به اسم قسم مفتاح USB. نغيره حسب الحرف المناسب. مثلاً، عندي هو N.

صورة توضح ما تم عمله لغاية الآن :

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\admin>cd c:\syslinux-3.71\win32

C:\syslinux-3.71\win32>syslinux -ma N:

C:\syslinux-3.71\win32>_
```

- نحمل الملف: initrdud.
ثم ننسخه في مسار مفتاح USB.
- نذهب إلى المسار:

```
E:\boot\i386\loader
```

- ثم ننسخ كل ما بداخل هذا المسار ونلصقه في مسار مفتاح USB.
- نفتح الملف isolinux.cfg، ثم نغير السطر الآتي، وباقي السطور القريبة منه (٣ أسطر في الأصل):

```
append initrd=initrd ramdisk_size=512000 ramdisk_blocksize=4096
splash=silent showopts
```

- نغير initrd=initrd لتصبح initrd=initrdud
- ونضيف الخيار kiwidebug

```
# openSUSE-11.0 [ ISO ]
label openSUSE-11.0 [ ISO ]
  kernel linux
  append initrd=initrdud kiwidebug=1 ramdisk_size=512000 ramdisk_blocksize=4096

# openSUSE-11.0 [ ISO ]
label Failsafe-openSUSE-11.0 [ ISO ]
  kernel linux
  append initrd=initrdud kiwidebug=1 ramdisk_size=512000 ramdisk_blocksize=4096

# mediacheck
label mediacheck
  kernel linux
  append initrd=initrdud kiwidebug=1 splash=silent mediacheck=1 showopts
```

- لا ننسى إعادة تسمية الملف من isolinux.cfg إلى syslinux.cfg.
- أخيرًا، كل ما علينا هو إعادة تشغيل الجهاز، واختيار الإقلاع من منفذ USB.

باستخدام نظام التشغيل "لينوكس"

- لنفترض أننا نريد تنصيب التوزيعة في مجلد المنزل الخاص بنا، نطبق الأوامر التالية:

```
$ su
```

نقوم بإدخال كلمة السر.

```
$ mkdir /media/iso
$ mount -o loop /home/bob-work/*openSUSE-11.0-* /media/iso
```

- نذهب إلى المسار التالي:

```
/media/iso/
```

- ثم نقوم بنسخ المحتوة ولصقه في مسار مفتاح USB.
- نطبق الأمر التالي، لمعرفة اسم جهاز مفتاح USB:

```
df -h
```

- قد يكون اسمه: sdc أو sdc١ أو sdd١ أو ما إلى ذلك. بعد معرفة اسمه نطبق الأمر التالي:

```
syslinux /dev/sdd
```

هذا على فرض أن اسم جهاز مفتاح USB هو sdd، غيره باسمه المناسب عندك.
- ننسخ محتوى المسار التالي:

```
boot/i386/loader
```

ثم نلصقه في مسار USB.
- نقوم بتحميل الملف initrdud ثم ننسخه في مسار مفتاح USB.
- نقوم بالتعديل على الملف isolinux.cfg كما شرحنا في الطريقة المتبعة مع نظام التشغيل "ويندوز"، ونجرب الإقلاع.
- لا ننسى عمل unmount للتوزيعة، لذلك نطبق الأمر التالي، بصلاحية الجذر:

```
umount /media/iso
```

والآن يمكنك التجول وتوزيعة اوبن سوزه في جيبك متى ما تشاء!





أداة MSEC لإدارة حماية النظام على "ماندريفا"

للكاتب: محمد الخياري

مقدمة



حزمة Mandriva-Security، والمعروفة باسمMSEC، كانت من الحزم الأساسية أو حزم القاعدة، إن صح التعبير، على Mandrake Linux من الإصدار ٧,٠، وقد تطورت هذه الأداة كثيرًا عمّا كانت عليه سابقًا؛ فقد تمت كتابتها كليًا بلغة "بايثون" على الإصدار ٨,٢.

يرجى الانتباه إلى أن هناك اختلاف بين إصدارات MSEC، وإن أغلب -وليس كل- ما سيتم ذكره في الموضوع ساري مفعوله على الإصدارات السابقة؛ مما سيؤدي إلى فشل بعض الأوامر على إصدارات MSEC الأقدم.
وظيفة MSEC كانت ولا تزال نفس الوظيف، وكل مستخدم، سواءً كان على علم أو لا، أن يكون استخدم هذه الأداة بأحد مستوياتها؛ فإن DrakX (أداة تثبيت النظام الرسومية على "ماندريفا")، في أحد مراحل التثبيت، يطلب من المستخدم تحديد مستوى الحماية المراد العمل به على النظام (ضعيفٌ جدًا، عاديّ، عالٍ، عالٍ جدًا، ...).

الولوج إلى واجهة MSEC الرسومية مع تحديد مستوى حماية النظام

قبل الدخول في تفاصيل الشرح يبدو أن كل قارئٍ لديه الرغبة في إلقاء نظرةٍ على واجهة MSEC الرسومية؛ فلنبدأ بهذه النقطة أولاً، وهذا لا يمنع من قراءة ما سيأتي لاحقًا طبعًا.

للولوج إلى الواجهة الرسومية لهذه الأداة نتبع التالي:

Menu > Outils > Outils système > Configurer votre ordinateur > Sécurité > Configurer le niveau de sécurité du système et l'audit de sécurité

أو نفتح سطر الأوامر، بصلاحيات الجذر root. بالطبع نكتب draksec وننقر على Enter.
مستوى الحماية الحالي لنظامك يظهر في أسفل النافذة الخاصة بالأداة على لائحةٍ متغيرة.

على كل قسم أو جزءٍ عدا Options de Base. يمكنكم اكتشاف قيم مختلف الخيارات الافتراضية الخاصة ب MSEC من أجل مستوى حماية نظامكم بالضغط على خانة Aide، أو Help بالإنجليزية.

الولوج من خلال الواجهة الرسومية إلى الصلاحيات الافتراضية المحددة من طرف MSEC

التصاريح المفروضة على الملفات والمسارات drakperm:
MSEC يفرض صلاحياتٍ محددةٍ على عددٍ من الملفات والمسارات داخل النظام.

للدخول إلى الواجهة التي تمكن من فحص، تعديل أو إكمال هذه الصلاحيات:
نفتح سطر الأوامر، ونستخدم صلاحيات المستخدم الجذر root، ثم نكتب الأمر drakperm.
أو بالواجهة الرسومية، من خلال:

Menu > Outils > Outils système > Configurer votre ordinateur > Sécurité > Ajuster finement les permissions du système

الصلاحيات المفروضة بواسطة النظام (أو بواسطة إعداداتكم المسبقة للحماية) ستظهر أمامكم داخل النافذة.

الصلاحيات الافتراضية لإنشاء الملفات umask:
لرؤية ماهي الصلاحيات الافتراضية الموزعة بواسطة MSEC أثناء إنشاء ملفumask؛ من أجل مستوى الحماية الحالي:

Menu > Outils > Outils système > Configurer votre ordinateur > Sécurité > Configurer le niveau de sécurité du système et l'audit de sécurité > Options système

نقوم بتمرير محتوى المساعدة Aide أو Help، ونتوقف عند:
Masque de permissions pour la création de fichier par les utilisateurs.
Masque de permissions pour la création de fichier par root.

للحصول على نفس المعلومات بواسطة سطر الأوامر؛ نستخدم صلاحيات المستخدم الجذر root، ثم ننفذ الأمر umask أو umask.

بعد ذلك، يمكننا تعديل الصلاحيات من خلال الواجهة (لا نقوم بأية خطوة إلا إذا كنا نعلم جيدًا ماذا نفعل).

بالطبع، وكما سنرى لاحقًا، فإن MSEC لا يتوقف عمله عند هذا الحد، بل يتعداه إلى أكثر من ذلك.

ما هو مستوى الحماية الأنسب؟
تاريخيًا، مستويات الحماية على MSEC كانت مُعرّفة بأسماءٍ خاصة، مثلاً المستوى ٠ يُسمى "BIENVENUE AU PIRATES"، الجملة تعني: "مرحبًا بالقراصنة"، وهكذا...

كيف أحدد المستوى الذي يناسبني؟
بالتأكيد توجد فكرةٌ وراء كل مستوى حماية، ويجب تحديد وتنسيق الحد بين الحماية وسهولة الاستخدام.

المستوى ٠: "مرحبًا بالقراصنة".
هذا المستوى يعتبر الأضعف، ويجب استخدامه بحذرٍ، ولا يتم اختياره أبدًا من خلال الواجهة الرسومية ل draksec، وهو يجعل النظام سهل الاستخدام، لكن على حساب الحماية بالطبع.

يمكنني طرح هذه الأسئلة، وإذا كان جواب أحدها "نعم"؛ فإن من الغباء أن أستخدم هذا المستوى من الحماية على نظامي:
هل جهازي متصلٌ بشبكة الإنترنت؟
هل جهازي متصلٌ بأجهزة أخرى عبر شبكة محلية؟
هل جهازي يستعمله مستخدمين آخرين غيري؟
هل لدي معلوماتٌ خاصةٌ وسريةٌ على جهازي؟

المستوى ١: "ضعيفٌ جدًا".
الفرق بين المستوى ١ و ٠: هو أن المستوى ١ يمكنك من استخدام أسماء و كلمات سرية؛ مما يجعل النظام قابلاً للاستخدام من قبل مستخدمين آخرين، لكن الخطر يبقى قائمًا إذا كان الجهاز على شبكة (إنترنت أو محلية).

المستوى ٢: "معياري".
الفرق بين المستوى ٢ و ١: هو أن MSEC يُرسل تنبيهات وتحذيرات مسبقة إن تم اكتشاف تحركات غير عادية، ويقوم بمراقبة أفضل.
هذا المستوى مثاليٌ لنظامٍ يستخدمه أكثر من مستخدمٍ، وهو الافتراضي في أغلب توزيعات لينوكس.

المستوى ٣: "عالي".
هو المستوى الأقل المنصوح به للأجهزة المتصلة بشبكة الإنترنت والشبكات المحلية. أغلب أساليب مراقبة الحماية تعمل على هذا المستوى، كمراقبة المنافذ المفتوحة مثلاً، ومع ذلك، فإن المنافذ المفتوحة تبقى مفتوحةً، والدخول من خلالها مسموحًا به؛ إذًا فهذا المستوى لا يناسب الأجهزة المتصلة بالإنترنت بغرض الخدمات (خادمات الويب، ftb، ssh). هذا المستوى يعطي قاعدةً جيدةً إذا أردت حماية نظامك بتعديل ملفات الإعدادت لمختلف الخدمات المتاحة.

المستوى ٤: " عالي جدًا".
وهو المستوى الذي يُنصح به للأجهزة الخادمة و الأنظمة المتصلة بالشبكة على الدوام. هذا المستوى يتيح الاتصال عن بعد بخادمات محددة وبجميع الوصلات المحلية. افتراضيًا، بعض الخدمات على هذا المستوى تكون غير مفعلة؛ إذًا، يجب على المستخدم تفعيلها يدويا وبصلاحيات المستخدم الجذر root. المستوى الرابع من الحماية يعطي ل MSEC شحنةً أكبر لمراقبة النظام وحماية المنافذ المفتوحة.

المستوى ٥: " المذعور"، بالإنجليزية "paranoid" أو "Paranoïaque" بالفرنسية.
أعلى مستوى من الحماية بحيث يقفل النظام بأكلمه. كل شيءٍ مراقبٌ بقوةٍ، وعلى المستخدم أن يقوم يدويًا بفتح المنافذ للسماح بعمل بعض الخدمات.

تعديل مستوى حماية النظام

تغيير مستوى الحماية على "ماندريفا" سهلٌ جدًا، سواءً من خلال سطر الأوامر، أو من خلال الواجهة الرسومية.

من خلال سطر الأوامر:
الشيء الوحيد الذي سنقوم به هو تشغيل MSEC، ثم نطلب منه تفعيل مستوى الحماية الذي نريد، أو الذي يناسبنا إن صح التعبير، وهذا ممكن بتطبيق الأمر التالي، بصلاحية المستخدم الجذر:

```
[root@ordi ~]# msec 2
```



هنا، قمت بتفعيل المستوى ٢: "معياري"؛ لحماية النظام. الأمر سهلٌ جدًا بالطبع.
للتذكير، هذا سرد للقيم المتاحة، والمستوى الذي يقابلها:
٠: مرحبًا بالقراصنة
١: ضعيفٌ جدًا
٢: عالي
٣: عالي جدًا
٤: معياري
٥: المذعور
يمكنكم أيضًا جمع هذه المعلومات في ملفٍ خاصٍ؛ لفحصها بعد ذلك حسب الرغبة بواسطة الأمر:

```
[root@ordi ~]# msec -o log=stderr 4 2> 3_to_4.msec
```

من خلال الواجهة الرسومية:
نتجه إلى:
Menu > Outils > Outils système > Configurer votre ordinateur > Sécurité > Configurer le niveau de sécurité du système et l'audit de l'ordinateur
سنجد قائمةً متغيرةً باسم niveau de securité، نختار مستوى الحماية المناسب، ثم ننقر زر الموافقة.
ولأخذ فكرةٍ عما آل إليه اختيارنا؛ يجب أن نعلم أن جميع الخيارات المفعلة والتي تعود إلى مستوى الحماية المختار، نجدها في الخانات التالية:
Options réseau
Options système
Vérifications périodiques

تعديل الخيارات من خلال الواجهة الرسومية بدون تغيير مستوى الحماية



لتخصيص MSEC، خيارنا الوحيد ليس فقط المرور من مستوى حماية إلى آخر؛ فإننا نستطيع، تمامًا، تعديل خيار واحد أو عدة خيارات بدون تغيير المستوى، مما يؤدي إلى إنشاء مستوى حماية خاصٍ بنا.

يمكننا فعل هذا بطريقةٍ سهلةٍ من خلال الواجهة الرسومية:

Menu > Outils > Outils système > Configurer votre ordinateur > Sécurité > Configurer le niveau de sécurité du système et l'audit de l'ordinateur

أو من خلال سطر الأوامر:
بتطبيق الأمر draksec، بصلاحيات المستخدم الجذر.

مداخل وأوامر التحكم في MSEC

ملفات الإعدادات:

خيارات المراقبة الدورية لحملية النظام نجدها في كل من:

/etc/sysconfig/msec
/var/lib/msec/security.conf

كل مرة نقوم فيها بتعديل المستوى الحالي للحماية، يقوم MSEC بتدوينها مع القيم الافتراضية.

للعلم؛ فإن مجموع الخيارات الافتراضية يتم تخزينها لمستويات الحماية الستة داخل ستة ملفات:

/usr/share/msec/level.[0-6]

بالإضافة إلى ذلك؛ فإن الصلاحيات الافتراضية على بعض المسارات، والتي تُعطى من قبل مستويات الحماية نجدها مخزنةً داخل:

/usr/share/msec/perm.[0-6]

الملف:

/etc/sysconfig/msec

يتم استخدامه في في عددٍ من سكريبتات الشيل Shell Scripts (ليست لدي خبرةٌ في البرمجة، لكن كملاحظاتٍ فقط)؛ فمثلاً تتم المناداة عليه من طرف:

/etc/profile.d/msec.sh
/etc/bashrc
/etc/profile

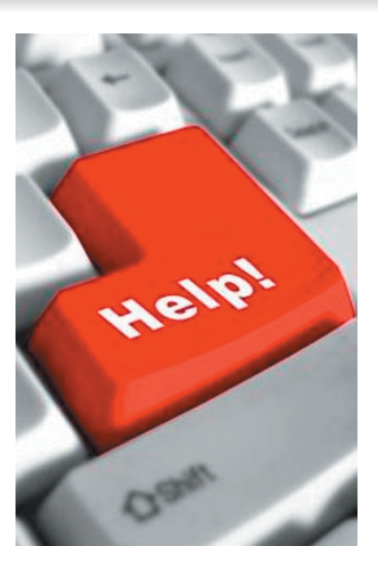
كل هذه الملفات هي ملفات إعدادات الصدفة Shell، في حين أن الملفات

/lib/msec/security.conf
/etc/security/msec/security.conf

يتم استخدامهما في المراقبة اليومية CHECK_SECURITY.

المتغيرات:

سنلقي نظرةً عن قربٍ لما يتيحه كل متغير خاصٍ بإعدادات MSEC، وهذه المتغيرات يتم تفعيلها حسب مستويات الحماية المعمول بها داخل النظام.



:CHECK_PASSWD
إذا كان مفعلاً؛ يقوم MSEC بالتأكد من أن جميع المستخدمين يملكون كلماتٍ سريةٍ، وأن هذه الكلمات السرية قد تم تشفيرها.

:CHECK_PERMS
إذا كان مفعلاً؛ يقوم MSEC بالتأكد من صلاحيات الملفات داخل المجلدات الخاصة بالمستخدمين، ويضع تقريرًا خاصًا بذلك. للعلم فإن MSEC لا يقوم بتغيير الصلاحيات، لكن يحدد المشاكل الممكنة فقط، ويقوم بمراقبة:
الملفات التي تعود ملكيتها للمستخدم، والتي لا يجب أن تحصل على صلاحيات الكتابة لباقي المستخدمين، مثلاً:

.bashrc .bash_profile .bash_login .bash_logout .cshrc .emacs .exrc .forward .klogin .login .logout .profile .tcshrc .fvwmrc .inputrc .kshrc .nexrc .screenrc .ssh .ssh/config .ssh/authorized_keys .ssh/environment .ssh/known_hosts .ssh/rc .twmrc .xsession .xinitrc .Xdefaults

الملفات التي تعود ملكيتها للمستخدم والتي تمتلك صلاحيات القراءة فقط:

.netrc .rhosts .shosts .Xauthority .gnupg/secring.gpg .pgp/secring.pgp .ssh/identity .ssh/id_dsa .ssh/id_rsa .ssh/random_seed

المجلدات الخاصة بالمستخدمين.

:CHECK_PROMISC
إذا كان مفعلاً؛ يقوم MSEC بمراقبة جميع بطاقات الشبكة، والتأكد من أنها على الوضع _promiscuous_.
البطاقات على هذا الوضع لها القدرة على اعتراض جميع حزم بروتوكول الإنترنت IP التي يتم استقبالها، وتتضمن أيضًا: الحزم التي يتم إرسالها بطرقٍ غير مباشرةٍ، والتي عادةً ما تنتج عن البرامج من نوع PACKET SNIFFER.

:CHECK_SECURITY
إذا كان مفعلاً؛ يقوم MSEC بتنفيذ السكريبت:

```
/usr/share/msec/security_check.sh
```

مع الأخذ بعين الاعتبار كل المتغيرات من نوع CHECK_،كما يقوم بالتأكد من واختبار الكثير من الأمور.

:CHECK_SGID
إذا كان مفعلاً؛ يقوم MSEC بالتحقق من ومقارنة قيمة ال md5 للملفات التي بها sgid_ bit_ مع القيم التي سبق حسابها، وهذا يُمَكّن من معرفة هل تم تعديل ملف ذي sgid_ bit_ بالرغم من أن حجمه وتاريخه متطابقان.

:CHECK_SHADOW
إذا كان مفعلاً؛ يقوم MSEC بالتأكد من أن جميع المستخدمين يمتلكون كلماتٍ سريةٍ غير فارغةٍ، وهذا ما يسمى ب Control Integrity على الملف:

```
/etc/shadow
```

:CHECK_SUID_MD5
إذا كان مفعلاً؛ يقوم MSEC بالتحقق ومقارنة قيمة ال md5 للملفات
التي تتوفر على suid_ bit_ مع القيم التي سبق حسابها ، هذا يمكن من معرفة إذا تم تعديل ملف ذو suid_ bit_ بالرغم من أن حجمه وتاريخه متطابقان .

:CHECK_SUID_ROOT
إذا كان مفعلاً؛ يقوم MSECبوضع تقرير عن كل التغييرات التي تطرأ على الملفات ذات suid_ bit_ ، وهذا يمكن من معرفة إذا كان هناك ملف من هذا النوع قد ظهر داخل النظام ، أو تم حذفه من النظام .

:CHECK_WRITABLE
إذا كان مفعلاً؛ يقوم MSEC بالبحث عن الملفات ذات صلاحيات الكتابة لكل المستخدمين، مع وضع تقريرٍ بذلك.

:CHKROOTKIT_CHECK
إذا كان مفعلاً؛ يقوم MSEC بالبحث داخل النظام عن RootKits.

:MAIL_EMPTY_CONTENT
إذا كان مفعلاً؛ تقرير الحماية سيتم إرساله بالرغم من أنه فارغ، وهذا يُمَكّن من معرفة هل MSEC في أفضل حالاتهن وأنه لم يتم التخلص منه من قبل أحد المُخربين Crackers أو أحد البرامج الضارة.

:MAIL_USER
يتيح إرسال تقريرٍ يومي إلى أحد المستخدمين. إذا لم تكن هذه الخاصية مفعلة؛ فإن الرسالة يتم إرسالها مباشرة إلى المستخدم الجذر.
قيم هذا المتغير تعود على ما سيتم اختياره في ADMINISTRATEUR SECURITY داخل خانة OPTION DE BASE أو "خيارات القاعدة" بالعربية.

:MAIL_WARN
إذا كان مفعلاً؛ سيتم إرسال بريد تنبيهٍ إلى المستخدم المحدد في المتغير MAIL_USER.

:PERM_LEVEL
يتم استعماله لتحديد أي ملف سيستخدم لإعطاء صلاحيات المستخدمين والمجموعات.
إذا كان مفعلاً؛ فإن MSEC يستخدم الملف:

```
/usr/share/msec/perm.$PERM_LEVEL
```

وإذا لم يتم تفعيله؛ فإن البرنامج يكتفي بالمتغير SECURE_LEVEL.
ولإعدادٍ دقيقٍ جدًا للنظام يتم استخدام الملف

```
/etc/security/msec/perm.local
```

إذا تأكد تواجده بالطبع.



:SYSLOG_WARN
إذا كان مفعلاً؛ يقوم MSEC بكتابة تقريره أيضًا على syslog.

:RPM_CHECK
يقوم بالتأكد من الحزم التي تغيرت من Hibernal Mode أو "وضع السبات".
والمراقبة تشمل أيضًا إذا تم تعديل ملفٍ تابعٍ لإحدى الحزم على النظام.

نتائج مسابقة قسم نفحات رمضانية
في مجتمع لينوكس العربي


المسابقة الرمضانية لمجتمع لينوكس العربي

يسر إدارة مجتمع لينوكس العربي ممثلة بالجهاز الإداري والإشرافي تقديم أطيب أسمى آيات التهنئة والتبريك لأعضاءنا الفائزين بمسابقة قسم "نفحات رمضانية" والتي جرى عقدها ضمن فعاليات شهر رمضان المبارك أعاده الله علينا وعليكم بالخير والمغفرة والبركات.

كما نشكر جميع من ساهم وشارك لإنجاح هذه المسابقة من أعضائنا الكرام، فقد كانت نسبة الإقبال كبيرة جدا وحماسية، وكانت المنافسة شديدة وقوية أيضا وللأيام الأخيرة من المسابقة والشهر الفضيل.

ونود أن نغتنم هذه الفرصة لشكر جميع المتبرعين من أموالهم الخاصة لدعم وتشجيع مثل هذه المسابقات الثقافية التي من شأنها رفع مستوى المخزون الفكري والثقافي للأعضاء في مجالات تاريخنا الإسلامي والعربي، وفي مجالات الثقافة المختلفة.

سيتم صرف جوائز بقيمة ١٥٠ دولار أميركي مقسمة لثلاثة جوائز بقيمة ٥٠ دولار أميركي للجائزة الواحدة، وسيتم تسليمها للفائزين على شكل هدايا من Amazon.com تستخدم لشراء أي من المنتجات المتوفرة من خلالهم ولأصحاب المراكز الثلاثة الأولى والذين زودونا بمعلوماتهم وهم حسب الترتيب التالي:

الفائز بالمركز الأول : محمد أحمد عقل الحسيني - مصر - عضوية EXp1r3d

الفائزة بالمركز الثاني : آمنة يوسف معوضة - اليمن - عضوية amena

الفائز بالمركز الثالث : اشرف موفق الحديدي - العراق - عضوية IraqiMousl

نبارك للفائزين الثلاثة مرة أخرى ونشكر كل من شارك ولو لمجرد المشاركة في هذه المسابقة التي سنقيمها بمناسبة شهر رمضان الفضيل في كل عام إن شاء الله.

إدارة مجتمع لينوكس العربي.

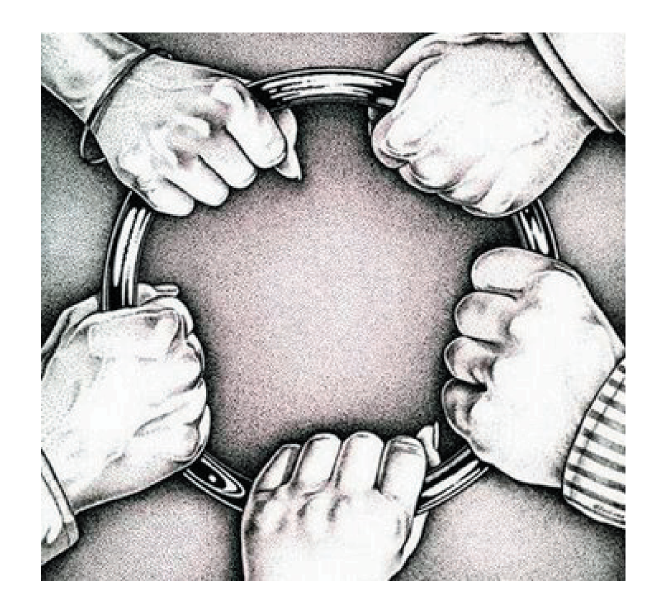


فريق عمل المجلة:

رئيس التحرير: سامر حداد GreyHunter

التدقيق اللغوي:

محمود سعيد محمود سعيد



هيئة التحرير:

مؤيد السعدي	alsadi
أنوار سيدام	Al AnWar
حنان العربي	أم عبد العزيز
روضة الصوابني	raoudha
أبو بكر الطليبي	bob-work
بدري دركوش	Free-Programmer
محمد عقل	EXp1r3d
محمد الخياري	knoppix_dark
أحمد عبدالرحمن	أحمد مصري
علي الشمري	B!n@ry

تصميم واخراج: سامر حداد GreyHunter

